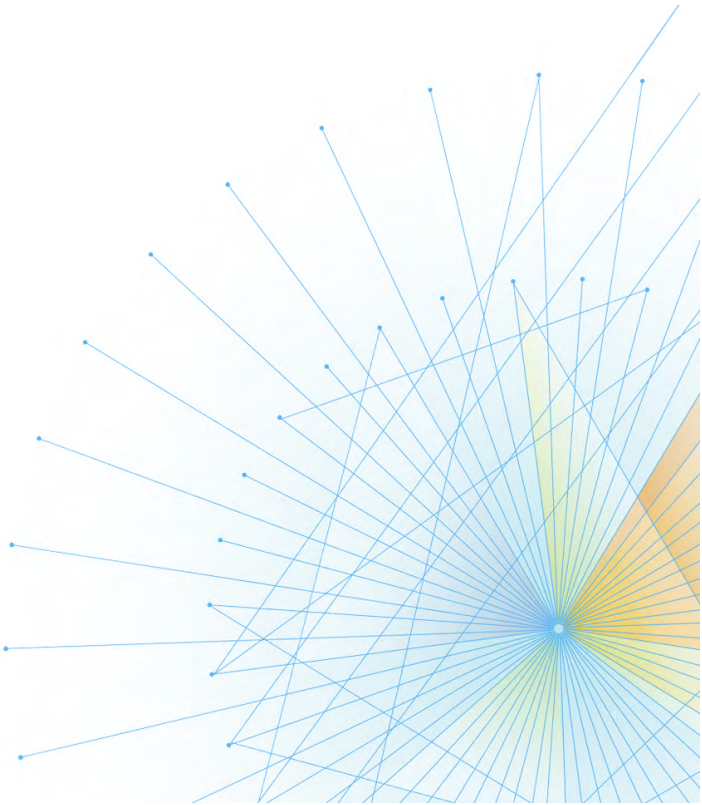
Compuware | The Mainframe Software Partner For The Next 50 Years

ISPW
Remote Server Guide

Release 17.02



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

Compuware Customer Support

https://go.compuware.com/

Doc. OCT2017
September 28, 2017

Contents

Introduction

This manual documents the ISPW Remote Server, and contains the following chapters and appendix:

- Chapter 1, “Overview”
- Chapter 2, “Remote Server Basics”
- Chapter 3, “Security (Authentication)”
- Chapter 4, “Remote Server – z/OS”
- Chapter 5, “Remote Server – Windows”
- Chapter 6, “Remote Server – Linux/Unix”
- Appendix A, “Parameters File rsparms.ini”.

Related Publications

- *ISPW Release Notes:* Overview of release features, supported operating environments, customer support, 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 Deploy Reference*: 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.
- *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 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 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 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 FrontLine customer support website at **https://go.compuware.com** in the following electronic formats:

- Release Notes in HTML format
- Product manuals in PDF format
- Adobe PDF index file (PDX file).

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 Support

Compuware provides a variety of support resources to make it easy for you to find the information you need.

Compuware FrontLine Customer Support Website

You can access online information for Compuware products via our FrontLine customer support website at **https://go.compuware.com**.

Compuware FrontLine provides access to critical information about your Compuware products. You can review frequently asked questions, read or download documentation, access product fixes, or e-mail your questions or comments. The first time you access Compuware FrontLine, you are required to register and obtain a password. Registration is free.

Contacting Customer Support

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**.

Web

You can report issues via the Quick Link **Create & View Support Cases** on the Compuware FrontLine home page.

Note: Please report all high-priority issues by telephone.

Mail

Compuware Customer Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099

Corporate Website

To access Compuware's site on the Web, go to **http://www.compuware.com**.

The Compuware site provides a variety of product and support information.

Chapter 1. Overview

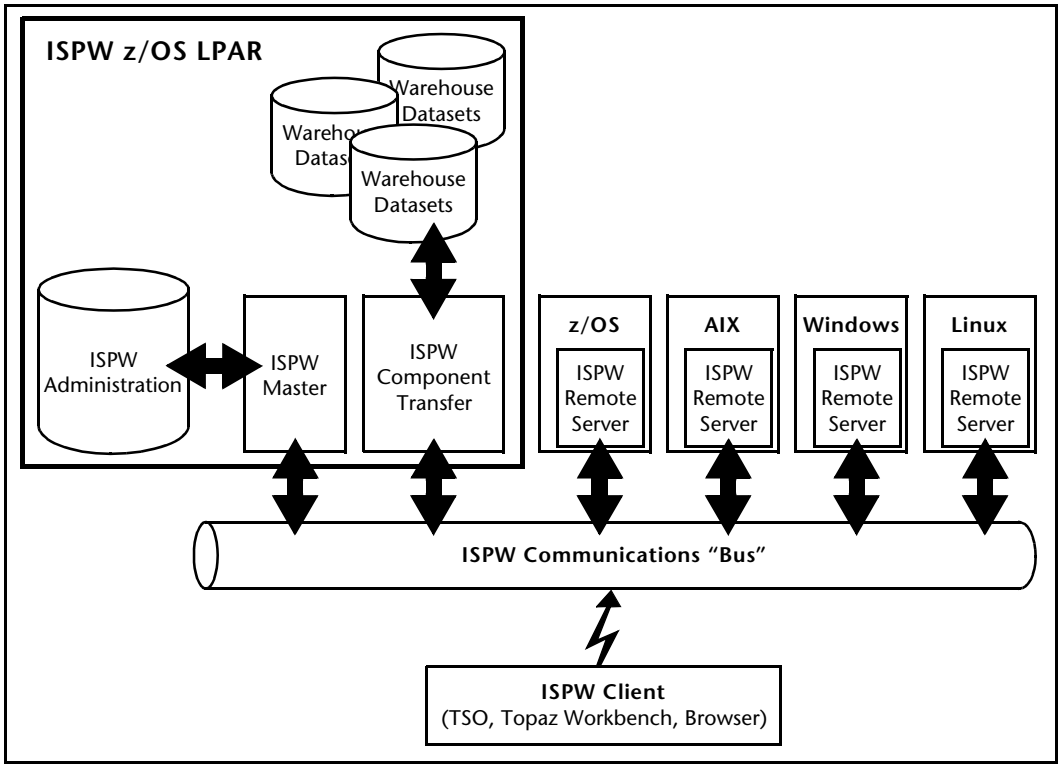
This manual describes the ISPW Remote Server. Remote Servers are ISPW Controlled Tasks running on platforms separate from the main ISPW Administration Server. ISPW Remote Servers are in constant communication with the ISPW Administration Server and are capable of:

- Receiving components to be stored in a local warehouse
- Receiving components to be stored on the local file system
- Accepting requests from the ISPW Master Server to initiate a deploy process
- Accepting Component Transport requests to retrieve a file from the local file system and send it to ISPWan ISPW Client
- Accepting Component Transport requests to receive a file from the requester and store it in the ISPW Warehouse or other location
- Accepting ComponenISPWt Transport requests to upgrade itself to a new version (Distributed servers only).

Architecture

Figure 1-1 illustrates the ISPW Architecture and where the Remote Servers fit in.

Figure 1-1. ISPW Architecture



Chapter 2.
Remote Server Basics

This chapter explains the fundamental concepts that apply to all ISPW Remote Servers, both z/OS Remote Servers and Distributed Remote Servers. It then describes those concepts that apply only to Distributed Remote Servers.

Concepts Applicable to All Platforms

This section describes the concepts and features that are applicable to Remote Servers on all platforms.

Common Design

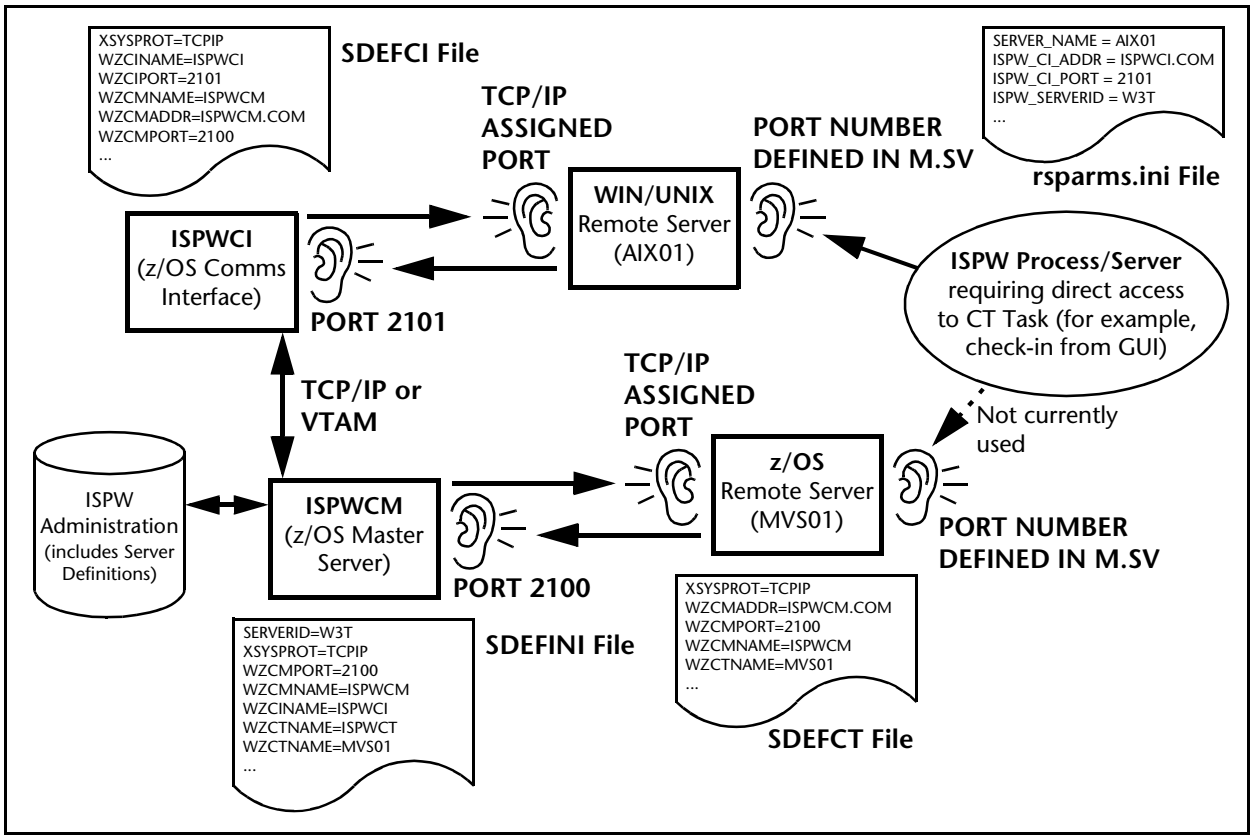
Remote Servers have the same functionality regardless of the platform on which they are executing. There are some technical differences between the z/OS Remote Server and the Distributed Remote Servers.

Method of Communication

Remote Servers establish a connection with the ISPW Master Server ISPWCM. This connection is done directly for z/OS and via ISPWCI for Windows, Linux, and Unix. Remote Servers also accept requests directly from ISPW Clients.

Figure 2-1 depicts the various Remote Server connections with a distributed parameters file called rsparms.ini.

Figure 2-1. Remote Server Communication



Remote Server Startup Process

When a Remote Server starts, the processing described in Table 2-1 takes place.

Table 2-1. Remote Server Startup

Sequence	Description
1	Read the parmfile (SDEFCT for z/OS and rsparms.ini for Distributed Platforms).
2	Establish a connection and log onto ISPW using the values in the parmfile. This connection will remain open for requests that originate from the ISPW Master Server (ISPWCM).
3	Server definition parms (defined in M.SV) are returned, which include the port number to listen on for requests coming directly from other ISPW processes, including ISPW CT and Remote Servers. Distributed Servers can also retrieve override parms stored in Extension Data against the M.SV entry (described in "Parms Stored in Extension Data" on page 2-5).
4	Listen on port defined in M.SV for remote requests. (Note that for Remote Servers on Distributed Platforms, there are currently no processes that contact this port, so effectively it is not used).

Local Warehouse

Each Remote Server has the capability of storing components in a local warehouse. ISPW's Deploy feature uses the warehouse to stage components prior to them being implemented on the local file system. On z/OS the warehouse is a PDSE implementation. On the other platforms, the warehouse copies are stored in a specific directory/path.

Local File System

Remote Servers have the capability of storing and retrieving components from the local file system. All requests for such access are directed via ISPWCM, where they are authorized according to the security rules in place.

Spawning Processes

Remote Servers have the ability to spawn processes for Deploy Operations. These processes run independently of the Remote Server and communicate back to CM the status of the implementation or activation. Table 2-2 lists the mechanisms by which the processes are spawned.

Table 2-2. Spawning Process

Platform	Mechanism
z/OS	Started Task. Remote Server issues the MVS Start command. The task will run under the authority of the UserID associated with the Started Task.
Windows	The Windows CreateProcess function is used to start a child process. The spawned process will inherit the security privileges of the remote server. The spawned process may be a program or BAT file, which must be located in the directory specified by the COMMAND_PATH in the Remote Server parm file.
Linux/Unix	The fork and execve functions of Linux/Unix are used to start a child process. The spawned process will inherit the security privileges of the remote server. The spawned process may be a program or shell script, which must be located in the directory specified by the COMMAND_PATH in the Remote Server parm file.

M.SV Entry

The ISPW Maintenance option M.SV is used to define Remote Servers. All Remote Servers must be defined here, regardless of the platform that they are running on. The M.SV entry screen is shown in Figure 2-2.

Figure 2-2. M.SV Entry Screen

```
ISPW              ADD SERVER TABLE DETAIL (W3M)
 Command ===>

 Enter required details:

  Server Name (KEY) ==>           (For LU6.2 must be VTAM Name)
  System Type       ==>           (MVS/WIN/AIX/LINX/SUN/HPUX)
  Server Type       ==>           (CT/RS/CM)
  Address           ==>
  Socket            ==>
  Deploy            ==>           (Y/N)
  Description       ==>

  Remote Status     ==> Offline


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

See the *ISPW Technical Reference Guide* for a description of the input fields. Note that the Address and Socket should be specified for z/OS Remote Servers, and only the Address is required for Distributed Remote Servers.

Concepts Applicable to Distributed Platforms

This section describes the concepts and features that are applicable to Remote Servers on the Distributed platforms. Individual chapters follow that describe specific detail for Windows, Linux, and Unix. See Chapter 4, “Remote Server – z/OS” for a description of the equivalent concepts for the z/OS platform.

Key Difference

Distributed Remote Servers are different from z/OS Remote Servers in that the management of the ISPW Software on a remote piece of hardware is more difficult than managing it on z/OS. This is mainly because there could be many hundreds of remote servers running on different operating systems in diverse locations across disparate time zones. Because of this, ISPW provides a Remote Server Monitor to help manage the Remote Servers by providing automation and connection management.

Server Status

In ISPW 4.4, the M.SV Panel was enhanced to show the status of Remote Servers. You will no longer have to select the specific Remote Server to see its status. As shown in Figure 2-3, the Remote Server list will show the status, the version, the ISPW/CI task it is connected to, and a count of inbound and outbound requests.

Figure 2-3. Server Status

```
ISPW              SERVER NAME TABLE (W3M)                        UPDATE MODE
Command  ===>                                                 Scroll ===> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, R Reset Token, X Extensions
               L View Log, T View Trace, P View Parms, Z Server Control

   Server     Type Stat CI Name  Version  ReqI  ReqO   Description
   _______    ____ ____ ________ ________ _____ _____  ____________________
_  AIX01      AIX  DISC                                 Remote Server on Mercur
_  BTA01      WIN  ACTV WZCIW3M  V4R00L40    62    14  Craig's Laptop
_  BTA02      WIN  ACTV WZCIW3M  V4R00L40   242     8  Craig's New Laptop
_  BTE01      WIN  DISC                                 Rustington
_  GABXP01    WIN  DISC                                 Gabriola Remote Server
_  HPX01      HPUX DISC                                 HP UNIX
_  ISPWSWE1   WIN  DISC                                 Stockholm
_  ITL01      WIN  DISC WZCIW3M               0     0  California
_  ITL02      WIN  DISC                                 Vancouver
_  JONES01    WIN  DISC                                 Steve Jones test RS
_  LNX01      LINX DISC                                 Remote Server on Pluto
_  PJSVR01    WIN  DISC WZCIW3M               0     1  Stockholm
_  PJSVR02    WIN  ACTV WZCIW3M  V4R00L37     1     0  Peter Laptop, Probably
_  ROBHW3M1   LINX DISC                                 Remote Server under lin
_  SMITH      WIN  DISC                                 Paul Smith Windows Lapt
_  SOL01      SUN  DISC                                 Sun Solaris
_  SWEWIN1    WIN  DISC                                 Stockholm
_  WIN01      WIN  ACTV WZCIW3M  V4R00L37  3776     1  Remote Server on Mars
_  WZCTW3M    MVS                                       ct for demo system
_  WZCTW3MR   MVS                                       remote ct
_  WZCTW3M6   MVS                                       ct on z16
_  WZCTW3T1   WIN  DISC                                 Gabriola Remote Server
_  WZCTZ113   MVS                                       ct on z/os 1.13
_  WZCTZ19    MVS                                       ct on z/os 1.9
_  W3M        MVS                                       ISPW/CM
****************************** Bottom of Data ******************************
```

Remote Server Monitor

The Remote Server Monitor (or Remote Monitor, for short) is a second ISPW executable which works in conjunction with the Remote Server to provide a more manageable ISPW Server.

Enabling the Remote Monitor

The Remote Monitor is enabled by setting a parameter in the rsparms.ini file. Once this is enabled, the Remote Monitor is started automatically when the Remote Server starts.

Monitor Functions

The activities listed in Table 2-3 will occur if the Remote Server Monitor function is enabled.

Table 2-3. Monitor Functions

Sequence	Activity
1	If the Remote Server Monitor detects the Remote Server is not running, it will start it
2	If the Remote Server detects the Remote Server Monitor is not running, it will start it.
3	If the Remote Server Monitor detects the Remote Server has lost its connection to the central server and cannot reconnect, it will restart the Remote Server.
4	If there is a scheduled outage window defined for the Remote Server (described in "Scheduled Outages" on page 2-9), the Remote Server Monitor will stop the Remote Server and keep it stopped until no longer in any outage window. If the Remote Server is started manually during the outage window, the Remote Server Monitor will shut it down.
5	If a file called shutdown.req is copied to the WORKING_PATH of the Remote Server, both the Remote Server and Remote Server Monitor will shut down. The shutdown.req file will be deleted by the Remote Server Monitor prior to shutdown.
6	If a file called maintmode.req is copied to the WORKING_PATH of the Remote Server, both the Remote Server and Remote Server Monitor will shut down. The maintmode.req file will ***not*** be deleted by the Remote Server Monitor prior to shutdown. The Remote Server and Remote Server Monitor will not be able to be started up again until the maintmode.req file is deleted from the working path.
7	The Remote Server and Remote Server Monitor can upgrade themselves to any version specified centrally in extension data. This feature is described in "Remote Server Automatic Upgrade" on page 2-13.

Controlling the Remote Servers via Deployment

The functions numbered 5 through 7 in Table 2-3 can be performed using ISPW's own Deploy function. This means that the upgrade of remote server software can be automated by simply setting up a special Deployment to send the software and control file to any or all Servers.

Parameters – rsparms.ini

Every Remote Server reads a configuration file on startup that contains the parameters required for its operation. The file name must be rsparms.ini, and it must reside in the working directory for the program startup. The parameters are described in Table 2-4, and the **Required** parameters are highlighted and appear near the top of the table.

Parms Stored in Extension Data

For ISPW 4.4 and above, on Server Startup the rsparms.ini parameters are copied to the Extension Data Classes WRSPARM and WRMPARM, where they are then managed centrally. (See Table 2-4.) Changes to the extension data values will override and replace

any parms in rsparms.ini, although new parameters added to rsparms.ini are still copied to the extension data.

Table 2-4. Parameters

Parameter	Description
[ISPWRS]	This is a required keyword in the file to delimit the Remote Server Parms (see example below).
SERVER_NAME	Specifies the name of this server. An entry of the same name must be defined in the Server table in ISPW/CM. **Required.**
ISPW_CI_ADDR	Specifies the TCP/IP name or address of the ISPW/CI that this server will connect to. **Required.**
ISPW_SERVERID	Specifies the logical name of the ISPW/CM server. **Required.**
WORKING_PATH	Specifies the default path for all files used and created by this Remote Server. This path may not be shared by multiple Remote Servers. **Required.**
ISPW_CI_PORT	Specifies the TCP/IP socket that this server should use to connect to the ISPW/CI task. **Required.**
RUN_MONITOR	Specifies if the Remote Server Monitor will be used. If this is set to yes, then the Remote Server will expect the Remote Server Monitor process to be active. If it is not, it will try to start it. The default value is NO. On a Windows system, the Remote Server ***must*** be running as a service to use the Remote Server Monitor. The ISPW 4.4 and above features to control servers from a central location require this parameter to be set to YES. **Recommended.**
EFFECTIVE_GROUP	Specifies the Effective Group that the Remote Server will run under. Ignored for Windows Systems. **Optional.**
EFFECTIVE_USER	Specifies the Effective User that the Remote Server will run under. Ignored for Windows Systems. **Optional.**
ISPW_ALT_CI_ADDR	Specifies the TCP/IP name or address of the alternate ISPW/CI that this server will connect to if the primary is not available. **Optional.**
ISPW_ALT_CI_PORT	Specifies the TCP/IP socket of the alternate ISPW/CI that this server will connect to if the primary is not available. **Optional.**
COMMAND_PATH	Specifies the path to the implementation and activation programs and scripts used by ISPW Deploy. If this is not specified, the WORKING_PATH is used. **Optional.**
WAREHOUSE_PATH	Specifies the path that will contain all of the warehouse directories and files. If this is not specified, the WORKING_PATH is used. **Optional.**
WAREHOUSE_HOUSEKEEPING	Specifies the interval in minutes between housekeeping requests. If this is not specified, no housekeeping will be done. During housekeeping, the Remote Server will contact ISPW/CM and check for any local warehouse entries that may be deleted. **Optional.**
ACTIVATION_LOG_ RETENTION	Specifies the number of days that deploy activation logs will be retained. The activation logs are created in the same path as the audit logs. The default value is 7. **Optional.**
LOG_PATH	Specifies the path that will contain all of the log files. If this is not specified, the WORKING_PATH is used. **Optional.**
LOG_VERSIONS	Specifies the number of versions of log files that will be retained. The default value is 3. **Optional.**
TRACE_PATH	Specifies the path that will contain all of the trace files. If this is not specified, the WORKING_PATH is used. **Optional.**
TRACE_VERSIONS	Specifies the number of versions of trace files that will be retained. The default value is 3. **Optional.**

Table 2-4. Parameters (Continued)

Parameter	Description
GATEWAY	Specifies if this Remote Server will run a Client Gateway using named pipes. If this is enabled, ISPW Client programs will be able to connect to the ISPW Server through this Remote Server. The default value is NO. **Optional.**
RUN_AS_DAEMON	Specifies whether this Remote Server will run as a Unix/Linux daemon. This parm is only valid on Unix or Linux systems. It is ignored on Windows systems. Valid values are: • YES – run as a Daemon • NO – run in command line mode. The default value is NO. If the Remote Server is started under the root user, then this parm is ignored and the process will run as a daemon. **Optional.**
WAREHOUSE_DIRECTORY_MODE	Specifies the default mode used when creating the warehouse directory on a Unix/Linux System. **Optional.**
WAREHOUSE_FILE_MODE	Specifies the default mode used when creating warehouse files on a Unix/Linux System. **Optional.**
WRITE_DIRECTORY_MODE	Specifies the default mode used when creating directories on a Unix/Linux System. **Optional.**
WRITE_FILE_MODE	Specifies the default mode used when creating files on a Unix/Linux System. **Optional.**
RS_START RS_START_ARGS	Specifies the command used to start the Remote Server. This pair of parameters is required if RUN_MONITOR=YES and the Remote Server is running on a Unix/Linux system. These parms are ignored on Windows systems.
[ISPWRM]	This is a required keyword in the file to delimit the Remote Server Monitor Parms (see example below).
EFFECTIVE_GROUP	Specifies the Effective Group that the Remote Server Monitor will run under. Ignored for Windows systems. **Optional.**
EFFECTIVE_USER	Specifies the Effective User that he Remote Server Monitor will run under. Ignored for Windows systems. **Optional.**
LOG_PATH	Specifies the path that will contain all of the log files. If this is not specified, the WORKING_PATH is used. **Optional.**
LOG_VERSIONS	Specifies the number of versions of log files that will be retained. The default value is 3. **Optional.**
TRACE_PATH	Specifies the path that will contain all of the trace files. If this is not specified, the WORKING_PATH is used. **Optional.**
TRACE_VERSIONS	Specifies the number of versions of trace files that will be retained. The default value is 3. **Optional.**
RM_START RM_START_ARGS	Specifies the command used to start the Remote Server Monitor. This pair of parameters is required if RUN_MONITOR=YES and the Remote Server is running on a Unix/Linux system. These parms are ignored on Windows systems. **Optional.**
UPLOAD_WAREHOUSE	Specifies the name of the warehouse that the Remote Server log and trace files will be uploaded to. **Optional.**
UPLOAD_CT_NAME	Specifies the name of the ISPW/CT server the Remote Server Monitor will connect to when it uploads Remote Server log and trace files. This is required if UPLOAD_WAREHOUSE is coded.
UPLOAD_CT_ADDR UPLOAD_CT_PORT	Specifies the TCP/IP name or address and port of the ISPW/CT server the Remote Server Monitor will connect to when it uploads Remote Server log and trace files. This is required if UPLOAD_WAREHOUSE is coded.

Note that both the Remote Server and Remote Server Monitor share the rsparms.ini file and also share the WORKING_DIRECTORY.

Example

See Appendix A, “Parameters File rsparms.ini” for an example rsparms.ini file.

Parameters – Extension Data Overrides

Parameter File Overrides

In ISPW 4.4 and above, the management of the parameters has been changed to be centrally focused. This means that parameter changes can be done from the Extension Data Classes specified against the Server in M.SV.

Override Extension Data

Table 2-5 describes the extension data classes which can be defined and the associated values which can be overridden.

Table 2-5. Override Extension Data

Extension Class	Description
WRMPARM	Remote Server Monitor Parms
WRSPARM	Remote Server Parms
WRSSCHED	Remote Server scheduled outages (For more information, see “Scheduled Outages” on page 2-9.)
WRSVRCTL	Used for the Automatic Upgrade Feature (For more information, see “Remote Server Automatic Upgrade” on page 2-13.)

Defining Extension Classes

Extension Classes are defined in M.EC. The four classes should be defined (with no parms). Then the next step is to import the samples as described in the following section. **This is a required step.**

Import Sample Classes

Sample Class definitions are supplied in the ISPW Samplib as member names of the same name as the class. Import these into M.EC and refresh the server.

Classes Automatically Defined to Server

Since ISPW 4.4, the WRMPARM and WRSPARM Classes are automatically added to the Remote Server and can be seen in M.SV(X). This is done as the parameters are copied into the extension data so that the parms can then be centrally managed.

Management of Parameter Values

Since ISPW 4.4, the parameters are centrally managed from the extension data, and new parameters defined in the extension data are copied to the server’s rsparms.ini file which is stored locally on the server. Any parms added to rsparms.ini are copied to the extension data and vice versa.

Specifying the Overrides

Parm overrides can be made using the format:

```
Parm = value
```

An example is shown in Figure 2-4.

Figure 2-4. Specifying Overrides

```
ISPW 4.0            SERVER: BTA01 EXTN:WRSPARM            Row     1 of    10
Command  ===>                                               Scroll ===> CSR

   Keyword  Value
   _______  ____________________________________________________________
   PARM01   LOG VERSIONS = 7
   PARM02   TRACE VERSIONS = 7
   PARM03
   PARM04
   PARM05
   PARM06
   PARM07
   PARM08
   PARM09
   PARM10
------------------------------ Bottom of List -------------------------------
```

Deleting a Parameter

To delete a parameter from both the extension data and rsparms.ini file, a semicolon is placed in the first position (in the extension data). The next time the server is started (or bounced) the parm will be deleted from both rsparms.ini and the extension data.

When the Overrides Take Effect

Overrides to parms take effect at the following times:

- When the Remote Server is restarted locally
- Every 24 hours at midnight
- By initiating a manual restart (Bounce) for a server from the M.SV(Z) panel.

Scheduled Outages

Scheduled Outage windows for the Remote Server task can be specified. During a Scheduled Outage, the Remote Server Monitor will stop the Remote Server and restart it again when the outage window completes. During an outage, the Remote Server is not in contact with the ISPW CM Task and will not accept Deployment Requests.

Defining a Scheduled Outage

Scheduled Outages can be defined in the following ways:

- Using a file called rssched.ini in the Working Directory
- Specifying the outage in Extension Data for the Remote Server
- Specifying the outage in Extension Data for the CM Task. (This defines a GLOBAL Outage for all Remote Servers.)

All scheduled outage definitions defined as described above are honored.

rsschd.ini

This file is used to define scheduled outages. The Remote Server reads it on startup (and restart). The example in Figure 2-5 shows the format required.

Figure 2-5. rsschd.ini

```
; ------------------------------------------------------------------------
;                ISPW Remote Server Scheduled Outages
; ------------------------------------------------------------------------
;
;  One or more scheduled outages can be defined in this file. The Remote Server
;  Monitor will stop the Remote Server at the beginning of each outage and re-start
;  it at the end of the outage.
;
;  The format of each scheduled outage is:
;
;  [OUTAGE_01]
;    Each outage MUST have a UNIQUR and SEQUENTIAL 2 digit number. The numbers
;    MUST start at "01" and go up sequentially.
;
;  TYPE=WEEKLY/ONETIME
;
;  DAY=day of week       For TYPE=WEEKLY. Values are upper case SUNDAY, MONDAY,
;                        TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY.
;
;  DATE=yyyy/mm/dd       For TYPE=ONETIME
;
;  FROM=hh:mm            24 hour clock
;  TO=hh:mm              24 hour clock
;
;  Notes. Since times cannot span midnight, to create a scheduled outage that
;         spans midnight you need two adjacent outages on adjacent days
;
[OUTAGE_01]
;
;  The following will schedule an outage every Sunday morning from 1:00am-4:00am
;
TYPE=WEEKLY
DAY=SUNDAY
FROM="01:00"
TO="04:00"
;
[OUTAGE_02]
;
;  The following will schedule an outage On Aug 30, 2010 from 10:00am-10:30am
;
TYPE=ONETIME
DATE="2010/08/30"
FROM="10:00"
TO="10:30"
```

Using Extension Data to Define an Outage

Extension data against the Server Definition in M.SV can be used to specify outages. The Extension Class is WRSSCHED, and the example in Figure 2-6 shows the format required.

Figure 2-6. Defining an Outage

```
ISPW 4.0            SERVER: SMITH2 EXTN:WRSSCHED          Row     1 of    10
Command  ===>                                               Scroll ===> CSR

   Keyword  Value
   _______ ____________________________________________________________
   PARM01   TYPE=WEEKLY
   PARM02   DAY=Tuesday
   PARM03   From=10:15
   PARM04   to=10:30
   PARM05   type=weekly
   PARM06   day=friday
   PARM07   from=10:01
   PARM08   to=11:01
   PARM09
   PARM10
------------------------------ Bottom of List -------------------------------
```

Trace, Log, Error, and Work Files

Various files are created by the Remote Server and Monitor processes. This section describes them.

Trace and Log Files

The Remote Server creates Trace and Log files according to the trace and log options specified in the configuration file. Files are created in the specified directory of the name listed in Table 2-6.

Table 2-6. Trace and Log Files

Type	Filename
Remote Server Log Files	audit*nnn*.txt where *nnn* is a number. The file audit000.txt is always the current Log File. Audit001.txt is the previous file and so on for as many versions as are specified to be kept in the configuration file (LOG_VERSIONS parameter).
Remote Server Trace Files	trace*nnn*.txt where *nnn* is a number. The file trace000.txt is always the current Trace File. Trace001.txt is the previous file and so on for as many versions as are specified to be kept in the configuration file (TRACE_VERSIONS parameter).
Remote Server Monitor Trace Files	rmtrace*nnn*.txt where *nnn* is a number. The file rmtrace000.txt is always the current Trace File. Rmtrace001.txt is the previous file and so on for as many versions as are specified to be kept in the configuration file (TRACE_VERSIONS parameter).

New versions of the files listed in Table 2-6 are created when the Remote Server is started.

Viewing Parm, Log, and Trace Files

It is possible to view the Parameter, Remote Server Trace, and Remote Server Log files from an ISPW TSO client using functionality in Maintenance Option M.SV. This is a useful aid in troubleshooting problems, because it is not necessary to physically log onto the Remote Server. After selecting option M.SV, the screen shown in Figure 2-7 is displayed.

Figure 2-7. Viewing Parm, Log, and Trace Files

```
ISPW              SERVER NAME TABLE (INT)                      UPDATE MODE
Command  ===>                                               Scroll ===> CSR

List Commands: A Add Entry, L Locate Entry, B Browse Mode
Line Commands: S Select, D Delete, R Reset Token, X Extensions
               L View Log, T View Trace, P View Parms, Z Server Control

   Server     Type  Description

   BTA01      WIN   Compuware ISPW (Australia) Server 01
   BTEW3TW1   WIN   Compuware ISPW Europe Windows Server
   SMITH3     WIN   Dave Ferguson Windows 7
   WZCHW3TA   AIX   Chattanooga AIX 5.2
   WZCHW3TL   LINX  Chattanooga Linux RH 7.3
   WZCHW3TW   WIN   Chattanooga Windows NT
   WZCTGAB1   WIN   Paul Johnson Windows 2000 Server
   WZCTW3T    MVS   CT Server for MVS
   WZCTW3T1   WIN   Paul Smith Windows 2000 Server
   WZCTW3T2   MVS   Second MVS Server
   WZCTW3T3   LINX  Paul Smith Linux Red Hat Server
   WZCTW3T4   WIN   Paul Smith Windows XP Laptop
   WZCTW3T5   WIN   Dave Ferguson
```

Entering **L** or **T** next to the Server entry will present a list of files that are available for viewing. Entering **P** will display the Parameter file that was used as input to the Remote

Server (including any overrides). Entering **Z** next to a Server Entry will allow additional Remote Server control functions to be initiated.

Error Files

The files listed in Table 2-7 may contain detailed error messages if the Remote Server start process does not work correctly.

Table 2-7. Error Files

Type	Filename
wzzrs.err	Error file stored in the Working Directory which contains errors about the Remote Server startup
wzzrm.err	Error file stored in the Working Directory which contains errors about the Remote Server Monitor

Work Files

Various files of different names are used to manage the different processes, especially around the automated upgrade of the Remote Server software.

Remote Server Control

Selection of a Remote Server using option Z takes the user into the Remote Server Control function.

Remote Server Control Screen

Entering option Z against a Server in M.SV displays the screen shown in Figure 2-8.

Figure 2-8. Remote Server Control Screen

```
ISPW H.SV         SERVER BTA02 CONTROL (W3M)
Command  ===>

Server Control:

  P - Stop               - Stop the Remote Server
  B - Bounce             - Stop and Restart Remote Server
  W - Wake Up            - Wake Up the Deploy Process
  H - Housekeeping Start - Start the Housekeeping Process

Trace Control:

  The following options should only be performed under the direction of
  ISPW Technical Support staff.

  R - Trace Reset   - Reset Trace Options to Initialization Defaults
  M - Trace Modify  - Modify Trace Options with Trace Control Statement

  Trace Control Statement (Required for Option M)

  ____________________________________________________________

Press Enter to continue or End to return to previous menu.
```

P – Stop

This will shut down the Remote Server. Note that it can only be started from the system the Remote Server is on.

B – Bounce

This will stop and start the Remote Server. It is necessary to bounce a server if any of the Remote Server extension data has changed and needs to be picked up.

W – Wake Up

When Deployments occur, the Remote Server is automatically triggered to contact the ISPW CM Task for any deployments it must do. On rare occasions, this trigger might not occur successfully, so the W option makes it possible to manually trigger a Remote Server to contact the ISPW CM Task for any deployments it may have in the queue.

When a Remote Server is started or Bounced, it also performs this check.

H – Housekeeping Start

The Remote Server housekeeping task is a process that cleans up its warehouse entries. How often this runs is determined by the value of the parm WAREHOUSE_HOUSEKEEPING. It can be triggered manually using option H from this screen.

Trace Functions

Server Trace functions can be initiated to help investigate and/or resolve any Remote Server problems. The Trace Control functions should only be used as directed by ISPW Customer Support (see “Customer Support” on page vi).

Remote Server Display

Status Display

The basic attributes for the Remote Server are displayed when selecting the Server from M.SV, as shown in Figure 2-9.

Figure 2-9. Remote Server Display

```
ISPW               BROWSE SERVER TABLE DETAIL (INT)
Command ===>

 Server Name (KEY) ==> WZCTW3T1  (For LU6.2 must be VTAM Name)
 Server Type       ==> WIN       (MVS/WIN/AIX/LINX)
 Address           ==> 192.168.0.22
 Socket            ==> 2222
 Deploy            ==> Y         (Y/N)
 Description       ==> Paul Scott Windows 2000 Server

 Remote Status     ==> Offline

Press END to return
```

Remote Server Automatic Upgrade

Remote Servers and Remote Server Monitor Processes can automatically upgrade themselves, without having need to access the machine that the Remote Server is running on.

How This is Done

Each Remote Server has an internal Version Number in the format VnRnnLnn. For example, the Remote Server version this document covers up to is V17R02L02, and this is reflected in the Remote Server Log as shown in Figure 2-10.

Figure 2-10. Version Number

```
******************************** Top of Data ********************************
2013/07/12 00.00.00 I Log initialized
2013/07/12 00.00.00 I File=C:\ISPW\audit000.txt
2013/07/12 00.00.00 I Running ISPW Remote Server for Windows - Version 17.02.02
```

An Extension Class called WRSVRCTL is associated with the Server in M.SV, and this is used to specify the version of software the Remote Server Tasks should be running. When the Remote Server is Started or Bounced, it checks this extension data for the level of software it should be running. If the level is different, it connects to a CT Task, retrieves the new software from a Warehouse, and Upgrades itself.

A new version of the Remote Server Task software is delivered in ISPW Warehouse Member format (that is, in a PDSE member). This member is placed in a warehouse and using Extension Data associated with the Remote Server, the name of the CT Task that can deliver the new software is specified.

Specify CT Server

Either in the extension data override (WRSVRCTL) or in RSPARMS.ini, the following parm is required which specifies where the Remote Server can get any new versions.

```
UPGRADE_SERVER = WZCTW3M
```

Specify Desired Software level

The WRSVRCTL Extension Class is used to specify what level of software the Remote Server is meant to be running.

WRSVRCTL

The format of the extension class is as shown in Figure 2-11 and listed in Table 2-8.

Figure 2-11. WRSVRCTL

```
ISPW 4.3            SERVER: BTA02 EXTN:WRSVRCTL          Row     1 of     3
Command  ===>                                              Scroll ===> CSR

   Keyword  Value
   _______ ____________________________________________________________
   CURRVER  V4R00L40
   PRODVER  V4R00L40
------------------------------ Bottom of List -------------------------------
```

Table 2-8. WRSVRCTL

Parm	Description
CURRVER	This contains the current version being run on the Remote Server. This is set by the Remote Server automatically and does not need to be set by the user.
PRODVER	This contains the desired production version of the Remote Server. If the current version is not the same as the production version, an upgrade (or regression) to the production version will be triggered. The exception to this rule is described in TESTVER below.

Table 2-8. WRSVRCTL (Continued)

Parm	Description
TESTVER	This contains a version that the Remote Server may be running without triggering an upgrade. This permits testing of new versions of the Remote Server without triggering the upgrade feature to automatically regress it back to the production version.

Upgrade Process

When the Remote Server starts up, it will set the current version and get the values for the production and test version. If the current version is not the same as either the production or test version, an upgrade is triggered to convert the Remote Server to the production version.

The Remote Server will look for a parameter in the rsparms.ini called UPGRADE_SERVER. This will define the name of the ISPW/CT server that has the upgrade warehouse defined. If this is not defined in the rsparms.ini, you can either add it into the rsparms.ini, or add it to the WRSPARM extension data for the server in M.SV.

The ISPW/CT server defined in the UPGRADE_SERVER parameter, must have the UPGRADEWAREHOUSE parameter coded in its startup parameters. This parameter identifies the name of the upgrade warehouse PDSE that is provided with ISPW. The upgrade warehouse PDSE contains the files that make up each version for each platform that supports the Remote Server.

If the Remote Server has the UPGRADE_SERVER defined, it connects to that CT server to download the files that make up the new version. Once the files for the new version have been downloaded, the Remote Server will signal the Remote Server Monitor to complete the upgrade.

The Remote Server Monitor will signal the Remote Server to shut down. Once the Remote Server has been shut down, it will rename the executable for the Remote Server, then restart it. After the Remote Server is back up running the new version, it will signal the Remote Server Monitor to shut down, so the Remote Server can rename the executable and restart the Remote Server Monitor running the new version.

Chapter 3.
Security (Authentication)

This chapter describes how Remote Servers are authenticated to ISPW. See the relevant chapter for each platform for a description of local security requirements.

Overview

Authentication is the process by which a Remote Server establishes its credentials with the ISPW Master Server (ISPWCM) whereby the Master Server confirms that the Remote Server is who it says it is.

If the authentication process were to be compromised, it would be possible for unauthorized Remote Servers to establish contact with the Master Server and participate in component transport functions.

Centralized Authentication

All Remote Servers are authenticated centrally by the ISPW Master Server (ISPWCM). This is achieved by the combination of the security checks listed in Table 3-1.

Table 3-1. Centralized Authentication

Check	Security
1	A SAF UserID of the same name as the Server Name is defined in the security product (i.e. RACF, ACF2, TOP SECRET).
2	The UserID has authority to the ISPW Security rule SERVER LOGON.
3	The UserID has authority to the ISPW Security rule SERVER CTIDENT.
4	An entry exist in the ISPW Server Table (M.SV) for the server name.
5	The use of security tokens.

See the *ISPW Technical Reference* for a description of the various Security Rule definitions.

No Exposure to Passwords

The primary objective is to have a secure setup with no exposure to passwords being stored locally and being sent to the ISPW Master Server (ISPWCM). This is achieved via the use of security tokens that change on a regular basis.

Connecting the First Time

When a Server connects for the first time, and passes security checks one to four in Table 3-1, it receives a Security Token that is used in conjunction with subsequent authentication requests. This Token is stored in a registry file that is in the working directory.

Note: It is recommended that the Remote Server be started as soon as possible after it has been defined, so that it can initiate the Token process. This will prevent any rogue processes connecting to ISPW with that definition and authority.

Security Token

The security Token is randomly generated and changed periodically. It is stored in the ISPW database by the CM Task on z/OS. It is encrypted and stored in the Remote Server registry file on open systems. The encryption incorporates a hardware serial number of the machine that the Remote Server is running on. This prevents the registry file from being copied and used on another machine.

Using the Token

Every time the Remote Server makes a request to ISPW, it sends its Security Token. As well as verifying the UserID, ISPWCM will check that the Token matches the one it issued. If it does not match, ISPWCM will reject the request.

Storing the Token

The Token is required to be securely stored on both the remote platform and within ISPWCM. Table 3-2 describes where it is stored and the security requirements required.

Table 3-2. Storing the Token

ISPW Server	Security Considerations
z/OS Master Server (ISPWCM)	Tokens for all Servers are stored in the DB2 Repository. It is recommended that the DB2 repository be protected and not able to be viewed.
z/OS Remote Server	Authentication on z/OS is guaranteed in that it is not possible for another UserID to “spoof” the CT Address space. Tokens are only required for Remote Servers running on the Distributed platforms.
Windows Remote Server	Tokens are stored in a file located in the working directory along with the ini parms file. Both these files should be protected from alteration by unauthorized users. The Token is saved in an encrypted format and cannot be copied and used on another system for another Remote Server.
Unix/Linux Remote Servers	Same as Windows.

Refreshing the Token

The security token is changed at each server logon.

Reset Tokens

It may be possible that the Security Token needs to be reset because it is out of sync with the ISPW CM Task. This could happen, for example, if the Remote Server was moved to a different machine. There is a mechanism in M.SV to reset the Tokens for a server. The effect of this is as though the remote server is connecting for the very first time.

Note: It is recommended that the Remote Server be started as soon as possible after the Reset Token has been done, so that it can initiate the Token process. This will prevent any rogue processes connecting to ISPW with that definition and authority.

Internal Tickets

For requests made directly between Remote Servers, ISPWCM will ensure that only authorized requests are made via the use of an internal ticket. If one Remote Server needs to connect to another Remote Server, ISPWCM will authorize the request and send an internal ticket to both servers so that they can verify that the request is valid.

Encryption

The signon string that is sent from the Remote Server across the network is encrypted. All other message data and file data is not encrypted.

Chapter 4.
Remote Server – z/OS

This chapter describes the z/OS implementation of the Remote Server.

Execution Environment

The Server runs as an MVS Started Task.

Installation

See the *ISPW Installation and Configuration Guide* for details, and follow the instructions for the setup of a CT Started Task.

Entry in M.SV

Ensure that there is an entry in M.SV to reflect the new Remote Server.

Entry in SDEFINI

Every z/OS Remote Server must be reflected by an entry in the SDEFINI file as shown in Figure 4-1. This file is input to the ISPWCM Task and is required for communication and security reasons.

Figure 4-1. Entry in SDEFINI

```
..
   WZCTNAME = WZCTW3T
   WZCTNAME = WZCTW3T2
   WZCTNAME = WZCTW3T3
..
```

The name to the right reflects the name of the Remote Server defined both in M.SV and in the Remote Server's SDEFCT file.

Chapter 5.
Remote Server – Windows

This chapter describes the Windows implementation of the Remote Server.

Execution Environment

The Remote Server runs as a Windows Service. The service will be set up during the installation process. The Remote Server may also be started from a command line, however the normal method of operation would be to run it as a service.

Required Components

Table 5-1 describes the supplied software components.

Table 5-1. Required Components

Name	Description
isa_d.dll isc_d.dll wzzmpe.dll	These are the dlls that are required.
wzzrs.exe	Windows Service executable
wzzrm.exe	Remote Monitor executable
wzzrsc.exe	Windows command line executable
rsparms.ini	A parm file that contains the configuration parameters
wzzrs.reg	A registry file that contains Remote Server settings

Security

At install time, the Remote Server will be set up to run as a Windows service under the local system account. If you want to alter security, you may set up an account for the service and alter the service definition to run under that account. The account will require basic permissions to the directories used by the Remote Server for warehouses, logs, and program files.

The Remote Server Service will be associated to the new account. File access will be based on the permissions for the account for the service. Any permissions to copy to target directories must be set for that account.

Deploy activation processes spawned by the Remote Server will also run under this account, so any access required by the activation process must be granted to the Remote Server account.

Security Token

As part of the authentication process, a security token is stored on the local machine in the file wzzrs.reg. This security token is stored in an encrypted format to prevent it from being copied and used by an unauthorized system.

Entry in M.SV

Prior to the installation of the Remote Server, you must define it in the Server table in ISPW. Using the option M.SV, you can list the current server definitions. Follow the steps to create a server definition for the new Remote Server.

See "Concepts Applicable to Distributed Platforms" on page 2-4 for the Remote Server definitions.

Installation

The installation follows the standard Windows installation process. The package is in the form of a self-extracting install program. Simply double-click on the install package and follow the instructions in the installation wizard.

The process will install the product files, setup the ini parms and registry file, define the Remote Server as a Windows service, then start the service.

Starting the Remote Server

The installation process will set up the service to start automatically on boot-up. The Windows management console can be used to alter the service properties. The name of the service is ISPW Remote Server.

The Remote Server may also be run from a command prompt. To start it this way, change to the directory where the Remote Server executables were installed, then enter the following command:

```
wzzrsc [parmfile]
```

If not specified, the `[parmfile]` will default to rsparms.ini.

Post-Installation Customization

You may change configuration parameters after installation by editing the ini file rsparms.ini. Using a text editor, you can alter the values of the parameters, save the changes, then restart the Remote Server.

A description of each parameter is contained as comments in the ini file that was created at installation time. See "Concepts Applicable to Distributed Platforms" on page 2-4 for a complete description of these parameters.

Chapter 6.
Remote Server – Linux/Unix

This chapter describes the Linux and Unix implementation of the Remote Server.

Execution Environment

The Remote Server runs either as a Daemon or in command line mode. This is determined by the configuration parameter RUN_AS_DAEMON.

Required Components

Table 6-1 describes the supplied software components.

Table 6-1. Required Components

Name	Description
wzzrs	Executable – Remote Server
wzzrm	Executable – Remote Server Monitor
rsparms.ini	A parm file that contains the configuration parameters
wzzrs.reg	A registry file that contains Remote Server settings

Security

At installation time, a UserID will be created on the system that matches the ID of the ISPW Remote Server. The install program will automatically create the UserID and set up basic permissions to the directories used by the Remote Server for warehouses, logs, and program files.

If the Remote Server is started under the root uid, it will set the effective uid to the one that matches its ServerID. If the Remote Server is started under another uid, it will attempt to set its effective uid. If it cannot set its effective uid, it will terminate.

File access will be based on the permissions for the effective uid (normally the UserID matching the Remote ServerID). Any permissions to copy to target directories must be set for that UserID.

Processes that are spawned will be set with a real and effective uid of the Remote Server. No root authority will be passed to the spawned processes.

Security Token

As part of the authentication process, a security Token is stored on the local machine in the file wzzrs.reg. This security Token is stored in an encrypted format to prevent it from being copied and used by an unauthorized system.

Entry in M.SV

Prior to the installation of the Remote Server, you must define it in the Server table in ISPW. Using the option M.SV, you can list the current server definitions. Follow the steps to create a server definition for the new Remote Server.

See "Concepts Applicable to Distributed Platforms" on page 2-4 for the Remote Server definitions.

Installation

You should be logged in as root to perform the installation of the Remote Server. The installation program will set up a group called ISPW and a UserID matching the name of the Remote Server. It will also set the ownership and permissions for the files and directories created during the install. If you are not logged in as root, all files will be owned by the installing user, and the ISPW group and Remote Server UserID will not be set up.

Unpack the tar file into a temporary directory with the following command:

```
tar -xvf <package_name>.tar
```

Once you have unpacked the tar file into a temporary directory, run the install program by entering the following command:

```
./install
```

There will be a set of default configuration parameters presented. You can alter any of the parameters by entering the option number for that parameter. When you select an option number, a description of the parameter will be displayed along with a prompt to allow you to change it.

When you have specified the parameter settings as desired, you can select the option to complete the install.

Starting the Remote Server

The default installation process will set the Remote Server to run as a Daemon and be started as part of the normal system startup.

The Remote Server may also be run from a command prompt. To start it this way, change to the directory where the Remote Server executables were installed, then enter the following command:

```
wzzrs [parmfile]
```

If not specified, the `[parmfile]` will default to rsparms.ini.

Post-Installation Customization

You may change configuration parameters after installation by editing the ini file rsparms.ini. Using a text editor, you can alter the values of the parameters, save the changes, then restart the Remote Server.

A description of each parameter is contained as comments in the ini file that was created at installation time. See "Concepts Applicable to Distributed Platforms" on page 2-4 for a complete description of these parameters.

Appendix A.
Parameters File rsparms.ini

Figure 6-1 through Figure 6-5 show the contents of the rsparms.ini file.

Figure 6-1. rsparms.ini – Part 1

```
; ------------------------------------------------------------------------------
;                ISPW Remote Server Initialization File
; ------------------------------------------------------------------------------
;
[ISPWRS]
;
; -------- This section contain parameters used by the Remote Server ---------
;
; SERVER_NAME     Specifies the name of this server. This is required and
;                 must be unique. It must also be defined in the Server
;                 table in ISPW/CM.
;
  SERVER_NAME=WZCTW3T1
;
; EFFECTIVE_GROUP, EFFECTIVE_USER  Specify the effective group and user that
;                 the Remote Server will run under. These are optional values.
;                 These are ignored on Windows systems.
;
;  EFFECTIVE_GROUP = ispwrs
;  EFFECTIVE_USER = wzctw3t6
;
; ISPW_CI_ADDR    Specifies the TCP/IP name or address of the primary ISPW/CI
;                 that this server will connect to. This is required.
;
  ISPW_CI_ADDR=192.86.33.199
;
; ISPW_CI_PORT    Specifies the TCP/IP port that this server should use
;                 connect to the primary ISPW/CI task. This is required.
;
  ISPW_CI_PORT=2208
;
; ISPW_ALT_CI_ADDR Specifies the TCP/IP name or address of the alternate
;                 ISPW/CI that this server will connect to if the primary
;                 ISPW/CI is not available..
;
  ISPW_ALT_CI_ADDR=192.86.33.199
;
; ISPW_ALT_CI_PORT Specifies the TCP/IP port that this server should use
;                 connect to the alternate ISPW/CI task.
;
  ISPW_ALT_CI_PORT=2201
;
; ISPW_SERVERID   Specifies the logical name of the ISPW/CM server. This
;                 is required.
;
  ISPW_SERVERID=W3T
;
; WORKING_PATH    Specifies the default path for all files used and created
;                 by this remote server. This path may not be shared by
;                 multiple remote servers. This is required.
;
  WORKING_PATH=C:\ISPW
;
; COMMAND_PATH    Specifies the path to the implementation and activation
;                 programs and scripts used ISPW Deploy. If this is not
;                 specified, the WORKING_PATH is used.
;
; COMMAND_PATH=C:\ISPW\CMDS
;
```

Figure 6-2. rsparms.ini – Part 2

```
; WAREHOUSE_PATH  Specifies the path that will contain all of the warehouse
;                 directories and files. If this is not specified, the
;                 WORKING_PATH is used.
;
; WAREHOUSE_PATH=C:\ISPW\WAREHOUSE
;
; WAREHOUSE_HOUSEKEEPING  Specifies the interval in minutes between
;                 housekeeping requests. If this is not specified, then
;                 no housekeeping will be done. During housekeeping,
;                 the remote server will contact ISPW/CM and check for
;                 any local warehouse entries that may be deleted.
;
; WAREHOUSE_HOUSEKEEPING =60
;
; ACTIVATION_LOG_RETENTION  Specifies the number of days a deploy
;                 activation logs will be retained. The activation logs
;                 are created in the same path as the audit logs. The
;                 default value is 7.
;
; ACTIVATION_LOG_RETENTION=7
;
; LOG_PATH        Specifies the path that will contain all of the log
;                 files. If this is not specified, the WORKING_PATH is used.
;
; LOG_PATH=C:\ISPW\LOG
;
; LOG_VERSIONS    Specifies the number of versions of log files that will
;                 be retained. The default value is 3.
;
; LOG_VERSIONS=7
;
; TRACE_PATH      Specifies the path that will contain all of the trace
;                 files. If this is not specified, the WORKING_PATH is used.
;
; TRACE_PATH=C:\ISPW\TRACE
;
; TRACE_VERSIONS  Specifies the number of versions of trace files that will
;                 be retained. The default value is 3.
;
; TRACE_VERSIONS=7
;
; GATEWAY         Specifies if this remote server will run a Client Gateway
;                 using named pipes. If this is enabled, then ISPW Client
;                 programs will be able to connect to the ISPW Server
;                 through this Remote Server. The default value is NO
;
; GATEWAY=YES
;
; Default Modes   Specifies the default file and directory modes when
;                 creating files and directories on a Unix/Linux system.
;                 These parameters are optional.
;
; WAREHOUSE_DIRECTORY_MODE=644
; WAREHOUSE_FILE_MODE=120
; WRITE_DIRECTORY_MODE=202
; WRITE_FILE_MODE=201
;
; RUN_AS_DAEMON   Specifies if this remote server will run as a Unix/Linux
;                 daemon. This parms is only valid on Unix or Linux systems.
;                 It is ignored on Windows systems. The default value is NO.
;                 If the Remote Server is started under the root user, then
;                 this parm is ignored and the process will run as a daemon.
;
; RUN_AS_DAEMON=NO
;
; RUN_MONITOR     Specifies if the Remote Server Monitor will be used. If
;                 this is set to yes, then the Remote Server will expect the
;                 Remote Server Monitor process to be active. If it is not, it
;                 will try to start it. The default value is NO.
;
;                 On a Windows system, the Remote Server MUST be running as
;                 a service to use the monitor.
;
  RUN_MONITOR=YES
;
```

Figure 6-3. rsparms.ini – Part 3

```
; RS_START        Specifies the command used to start the Remote Server. This
;                 pair of parameters are required if RUN_MONITOR=YES and
;                 the Remote Server is running on a Unix/Linux system. These
;                 are ignored on Windows systems.
;
; RS_START_CMD=\u\ISPW\wzzrs
; RS_START_ARGS=rsparms.ini
;
; DEBUG/TRACE     The following set of parameters are related to tracing the
;                 Remote Server and should only be added under the direction
;                 of ISPW Customer Support. Default tracing options are:
;
;                 DEBUG = YES
;                 TRACE_GLOBAL = COMPID=ALL,TRTYPE=(1,2)
;
; DEBUG=YES
; TRACE_GLOBAL=COMPID=ALL,TRTYPE=(1,2,3)
; TRACE_GLOBAL=COMPID=ALL,TRTYPE=ALL
;
  TRACE_WZZRSINI=COMPID=ALL,TRTYPE=ALL
; TRACE_WZZRSTRM=COMPID=(5,6,17,18,19),TRTYPE=ALL
  TRACE_WZZRSPOP=COMPID=ALL,TRTYPE=ALL
; TRACE_WZZRSSQP=COMPID=(5,6,17,18,19),TRTYPE=ALL
; TRACE_WZZMSIMP=COMPID=(17,18),TRTYPE=(1,2,3,4)
  TRACE_WZZMSCMP=COMPID=ALL,TRTYPE=ALL
  TRACE_WZZCTLCM=COMPID=ALL,TRTYPE=ALL
; TRACE_WZZCTRAC=COMPID=(5,6,7,17,18,19),TRTYPE=(1,2,3,4,5,6,7,8,20,21,22)
; TRACE_WZZCTRAL=COMPID=(3,5,6,17,18),TRTYPE=ALL
; TRACE_WZZCTRAS=COMPID=(2,3,5,6,17,18),TRTYPE=ALL
  TRACE_WZZCDNWP=COMPID=ALL,TRTYPE=ALL
  TRACE_WZZCDPQP=COMPID=ALL,TRTYPE=ALL
; TRACE_WZZCWHKP=COMPID=(5,6,17,18),TRTYPE=ALL
; TRACE_WZZCXSTA=COMPID=(5,6,17,18),TRTYPE=ALL
;
[ISPWRM]
;
; ---- This section contain parameters used by the Remote Server Monitor -----

;
; EFFECTIVE_GROUP, EFFECTIVE_USER  Specify the effective group and user that
;                 the Remote Server Monitor will run under. These are optional
;                 values. These are ignored on Windows systems.
;
;  EFFECTIVE_GROUP = ispwrs
;  EFFECTIVE_USER = wzctw3t6
;
; LOG_PATH        Specifies the path that will contain all of the log
;                 files. If this is not specified, the WORKING_PATH is used.
;
; LOG_PATH=C:\ISPW\LOG
;
; LOG_VERSIONS    Specifies the number of versions of log files that will
;                 be retained. The default value is 3.
;
; LOG_VERSIONS=7
;
; TRACE_PATH      Specifies the path that will contain all of the trace
;                 files. If this is not specified, the WORKING_PATH is used.
;
; TRACE_PATH=C:\ISPW\TRACE
;
; TRACE_VERSIONS Specifies the number of versions of trace files that will
;                 be retained. The default value is 3.
;
; TRACE_VERSIONS=7
;
; RUN_AS_DAEMON   Specifies if this Remote Server Monitor will run as a Unix/Linux
;                 daemon. This parms is only valid on Unix or Linux systems.
;                 It is ignored on Windows systems. The default value is NO.
;                 If the Remote Server Monitor is started under the root user,
;                 this parm is ignored and the process will run as a daemon.
;
; RUN_AS_DAEMON=NO
;
```

Figure 6-4. rsparms.ini – Part 4

```
; RM_START        Specifies the command used to start the Remote Server Monitor.
;                 This pair of parameters are required if RUN_MONITOR=YES and
;                 the Remote Server is running on a Unix/Linux system. These
;                 are ignored on Windows systems.
;
; RM_START_CMD=\u\ISPW\wzzrm
; RM_START_ARGS=rsparms.ini
;
; UPLOAD_WAREHOUSE Specifies the name of the warehouse that the Remote Server
;                 log and trace files will be uploaded to. This is optional.
;
; UPLOAD_WAREHOUSE=WZCTW3T
;
; UPLOAD_CT_NAME  Specifies the name of the ISPW/CT server the Remote Server
;                 Monitor will connect when it uploads Remote Server log
;                 and trace files. This is required if UPLOAD_WAREHOUSE is
;                 coded.
;
; UPLOAD_CT_NAME=WZCTW3T
;
; UPLOAD_CT_ADDR  Specifies the TCP/IP name or address of the ISPW/CT server
;                 the Remote Server Monitor will connect when it uploads
;                 Remote Server log and trace files. This is required if
;                 UPLOAD_WAREHOUSE is coded.
;
; UPLOAD_CT_ADDR=192.86.33.199
;
; UPLOAD_CT_PORT  Specifies the TCP/IP port of the ISPW/CT server the Remote
;                 Server Monitor will connect when it uploads Remote Server log
;                 and trace files. This is required if UPLOAD_WAREHOUSE is
;                 coded.
;
; UPLOAD_CT_PORT=2211
;
; DEBUG/TRACE     The following set of parameters are related to tracing the
;                 Remote Monitor and should only be added under the direction
;                 of ISPW Customer Support. Default tracing options are:
;
;                 DEBUG = YES
;                 TRACE_GLOBAL = COMPID=ALL,TRTYPE=(1,2)
;
; DEBUG=YES
; TRACE_GLOBAL=COMPID=ALL,TRTYPE=(1,2,3)
; TRACE_GLOBAL=COMPID=ALL,TRTYPE=ALL
;
; TRACE_WZZRMINI=COMPID=ALL,TRTYPE=ALL
; TRACE_WZZRMTRM=COMPID=(5,6,17,18,19),TRTYPE=ALL
; TRACE_WZZRMPOP=COMPID=ALL,TRTYPE=ALL
; TRACE_WZZRMRAC=COMPID=ALL,TRTYPE=ALL
;
; ------------------------------------------------------------------------------
;
; TRACE USAGE NOTES:
;
; TRACE_procname = COMPID=(nn,nn,nn,...),TRTYPE=(nn,nn,nn,..)
; TRACE_procname = COMPID=ALL,TRTYPE=ALL
;
;  COMPID  ISPW USE  MPE USE
;  ------  --------  -------
;    2                ETS
;    3                ISA
;    5                BASE
;    6                ROUTER
;    7                ATTRIB
;    17     WZTR_MS   USER1
;    18     WZTR_CT   USER2
;    19     WZTR_RS   USER3
;    20               USER4
;
```

Figure 6-5. rsparms.ini – Part 5

```
;  TRTYPE  ISPW USE  MPE USE  LABEL
;  ------  --------  -------  -----
;    1                FORCE     X
;    2      WZTR_E    ERROR     E
;    3      WZTR_W    WARNING   W
;    4      WZTR_I    INFO      I
;    5                FLOW1     F1
;    6                FLOW2     F2
;    7                FLOW3     F3
;    8                FLOW4     F4
;    9                DATA1     D1
;    10               DATA2     D2
;    13               DIAG1     G1
;    14               DIAG2     G2
;    17               MEMUSE    M
;    18               RESUSE    R
;    19               PERF      P
;    20               ATTRBUF   AT
;    21     WZTR_D    USER1     U1
;    22     WZTR_T    USER2     U2
;
;  The format of the trace records are:
;
;  yyyy/mm/dd hh.mm.ss tttttttt ll Trace text...
;
;  tttttttt = the thread id.
;  ll       = the label (as described above).
;
; ------------------------------------------------------------------------------
```