

ARBUTUS zSERIES SERVER

INSTALLATION GUIDE

ARBUTUS SOFTWARE

Copyright

Copyright © 2017 Arbutus Software Inc. All rights reserved.

This manual may contain dated information. Use of these materials is based on the understanding that this manual may not contain all the information, or the most current information, relevant to an intended application. Support and update services are available from Arbutus, by separate written agreement.

Published January 2017 by:

Arbutus Software Inc.
270 - 6450 Roberts Street
Burnaby, British Columbia
Canada V5G 4E1

Telephone: +1-604-437-7873

Fax: +1-604-437-7872

E-mail: Info@ArbutusSoftware.com

Web: www.ArbutusSoftware.com

"Arbutus", "Analyzer", "Arbutus ConnectPlus" and "Smart Link for SAP ERP" are registered trademarks of Arbutus, or one of its subsidiaries or licensors. Use of a term in this manual should not be regarded as affecting the validity of any trade or service mark.

About this Guide

This guide is provided to licensed users of the Arbutus zSeries Server and will assist you with installing and configuring the Arbutus zSeries Server.

For more information about our products and services, visit our web site at:
www.ArbutusSoftware.com

Need Help?

Resources

For assistance in using the Arbutus zSeries Server, Analyzer or ConnectPlus, refer to these additional resources:

- Analyzer Installation Guide
- Analyzer Online Help
- ConnectPlus Installation Guide
- Arbutus zSeries Server Installation Guide

Contacting Us

If you have suggestions for improvements or if you can not find a solution to a problem, you can contact technical support for assistance. Please include:

- Your Analyzer serial number.
- A description of what you were doing when the problem occurred.
- The error message, if any, displayed by Analyzer.

Support Center

Global Help Desk

- Phone: +1.604.437.7873 or +1.877.333.6336 ext. 333
- Fax: +1.604.437.7872
- E-mail: support@ArbutusSoftware.com

Or write to us at:

Arbutus Software Inc.
270 - 6450 Roberts Street
Burnaby, BC
Canada V5G 4E1

Contents

Copyright	ii
About this Guide	iii
Contents	1
Chapter 1 Introduction	4
System Description	4
The Arbutus Solution - Overview	4
The Arbutus Solution - Components	5
User Interface Options	6
System Requirements	7
The Arbutus zSeries Server	7
Chapter 2 Installing the zSeries Server	8
Pre-Installation Considerations	8
Security	8
Start Options for the Session Controller	9
Session Processing Options	11
Licencing and Authorization	12
Upgrading an Existing System	12
Installing a New System	12
Copying Components from the Arbutus Website	12
Copying Components from Tape	14
Required Installation Procedures	17
Step 1a - Edit the Session Controller JCL for Spawned Jobs	18
Step 1b - Edit the Session Controller JCL for Sub-tasking	19
Step 2 - Edit the ASIINI Data Set	21
Step 3 - Edit the Session JCL	28

Optional Installation Procedures	29
Step 4 - Edit the IMS Session JCL	30
Step 5 - Establish Default DB2 Values	32
Step 6 - Establish the Default IMS Values	32
Step 7 - Establish Default Adabas Values	32
Step 8 - Allocate COBOL, PL/1 and EASYTRIEVE DBD Source Libraries	32
Start the Session Controller Job	32
Test PC-based Connections	33
Troubleshooting	33
Chapter 3 Optional zSeries Interfaces	35
The DB2 Interface	35
Using DB2 Version 9.1 ODBC	35
Creating the DB2 Plan	35
Specifying DB2 Subsystems	36
Additional Options for ODBC	36
Optional Uncommitted Read Clause	37
DB2 Table List	38
DB2 Security	39
The IMS Interface	40
The Arbutus zSeries Server Program Specification Block (PSB)	40
Updating the Arbutus zSeries Server ASIINI Data Set	41
The Adabas Interface	42
Updating the Arbutus zSeries Server ASIINI Data Set	43
Updating the Arbutus zSeries Server Session JCL	43
System Sort	43
System Sort DDNames	44

Sort Parameters	44
Chapter 4 zSeries Server System Messages	46
Chapter 5 Offline Interface	50
Overview	50
System Requirements	50
The ASIINI Data Set	50
Edit the Offline Execution JCL	51

CHAPTER 1 INTRODUCTION

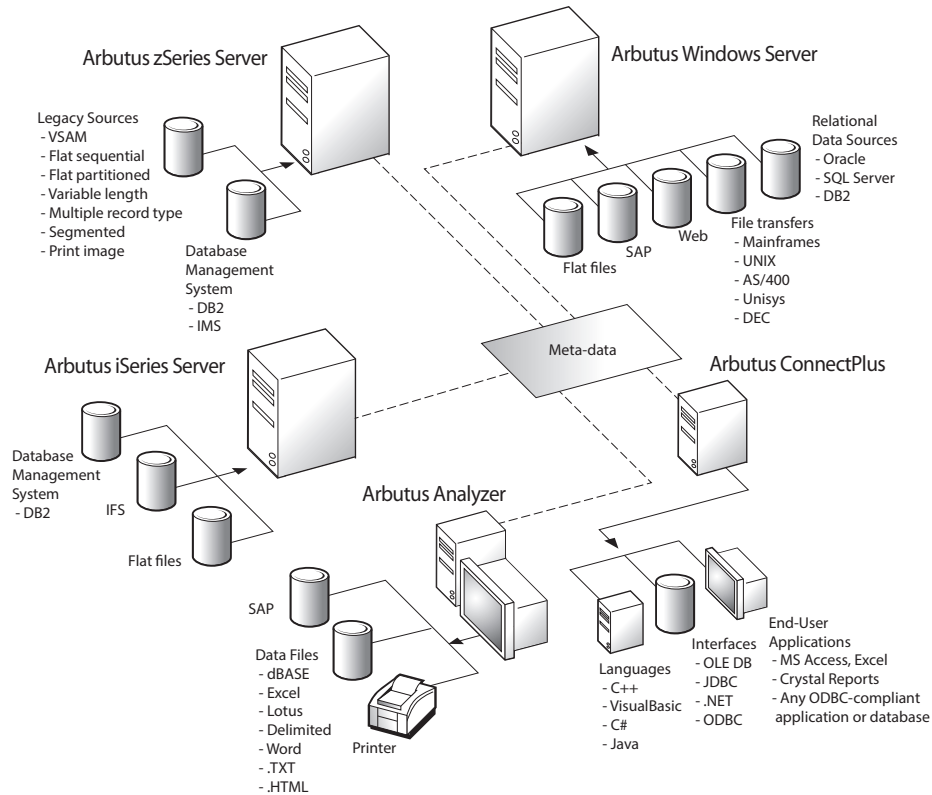
System Description

The Arbutus Solution - Overview

The Arbutus Solution consists of:

1. One or more of the Arbutus Servers
 - A zSeries mainframe computer running the Arbutus zSeries Server
 - An iSeries midrange computer running the Arbutus iSeries Server
 - A Windows server running the Arbutus Windows Server
2. One or more of the PC-based Arbutus Clients
 - Arbutus Analyzer
 - Arbutus ConnectPlus
3. A TCP/IP network connection linking the clients and servers together
4. Optional background offline batch processing for the Arbutus zSeries Server.

The Arbutus Solution



The Arbutus Solution - Components

An Arbutus “session controller” is provided to manage TCP/IP communications and Arbutus end-user sessions (“server sessions”) for configurations that connect across multiple platforms, as is the case when running the PC-based Arbutus Clients with any of the Arbutus Servers.

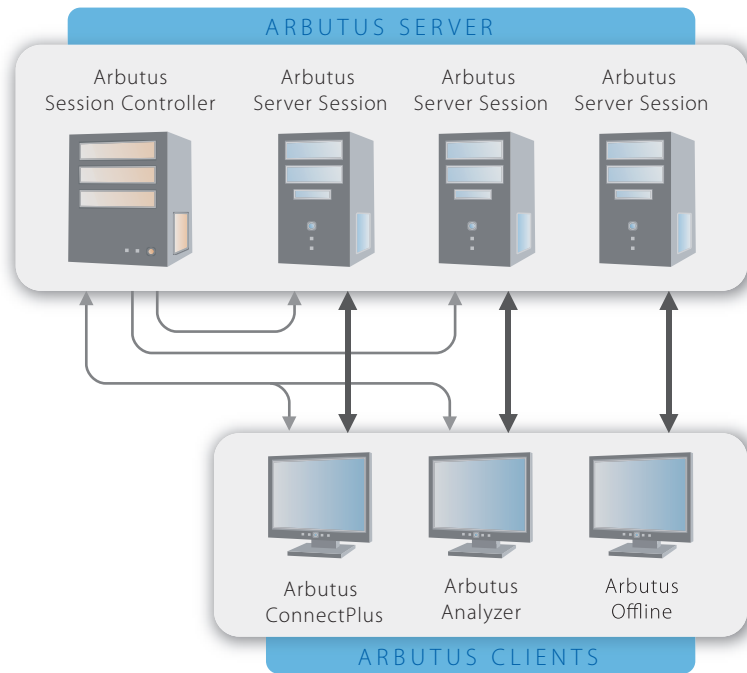
When a PC-based Arbutus Client sends a connection request to the session controller, the session controller authenticates the request and creates an end-user server session. The session controller then establishes a direct communications link between the client and the end-user server session, and hands off the communication link to them.

Once this connection is established, the end-user server session runs independently of the session controller, and the session controller goes back to listening for incoming requests.

For each connection it accepts, the session controller starts a separate end-user server session. Connection requests from different users result in multiple, independent server sessions. A server session ends when the connection is terminated from the PC-based client.

Note: Specific to the Arbutus zSeries Server, offline batch processing is also available. Offline batch processing does not require TCP/IP services and so does not require the services of the session controller (see diagram that follows). For more information see Appendix Chapter 5: “Offline Interface”.

Arbutus Server Session Management



User Interface Options

All of the Arbutus Servers can be configured to provide interactive, real-time processing from any ODBC-compliant PC application or from the Arbutus Analyzer graphical user interface.

In addition, the Arbutus zSeries Server can be configured to run in offline batch processing mode.

ODBC-Compliant PC Application

Use of this feature requires an ODBC-compliant application running on the PC, Arbutus ConnectPlus, TCP/IP services, various specific configuration settings, and at least one of the Arbutus Servers.

Arbutus Analyzer Graphical User Interface

Use of this feature requires a copy of the Arbutus Analyzer client running on the PC, TCP/IP services, various specific configuration settings and at least one of the Arbutus Servers.

Offline Batch Processing

Specific to the Arbutus zSeries Server, sessions can be run using the offline JCL provided.

Note: While the Arbutus zSeries Server installation process is essentially the same regardless of the user interface options selected, there are a few specific configuration differences between online and offline processing environments. For more information see Appendix Chapter 5: "Offline Interface".

System Requirements

The Arbutus zSeries Server

The installation steps described in "Installing the zSeries Server" on page 8 are those required to install the Arbutus zSeries Server for use in the client/server environment with either ConnectPlus or Analyzer.

Additional installation steps needed to support the offline batch interface are described in Appendix Chapter 5: "Offline Interface".

The Arbutus zSeries Server requires approximately 200 (3390) tracks of permanent disk storage. Virtual and temporary storage requirements vary depending on the functions performed.

The Arbutus zSeries Server can be configured to use either SyncSort, DFSORT, or CA-Sort, in addition to its own sort routine. Instructions for setting up Arbutus zSeries Server to use these sort utilities are included in "System Sort" on page 43.

The optional DB2 interface uses DB2 Version 5 (or later) DSNALI, DSNHLI2, and DSNTIAR service routines.

The optional IMS interface uses IMS Version 2 Release 1 (or later) interface modules.

The optional ADABAS interface reads ADABAS Version 6 and later data.

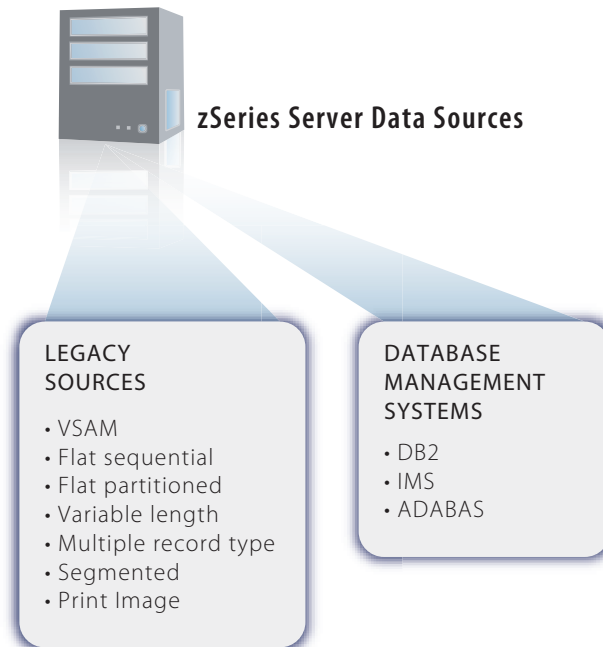
Additional installation steps needed to support the optional DB2 and IMS interfaces are described in "Optional zSeries Interfaces" on page 35.

Access via Analyzer or ConnectPlus uses the TCP/IP communication protocol, Version 3 Release 1 (or later).

CHAPTER 2 INSTALLING THE zSERIES SERVER

Installing the Arbutus zSeries Server is quick and easy. Here you will learn about copying Arbutus zSeries Server components from tape or the Arbutus Website, the configuration options, and performing the Arbutus zSeries Server install.

The following diagram details the data sources available to the Arbutus zSeries Server:



Pre-Installation Considerations

This section describes the options you have for installing the Arbutus zSeries Server and customizing the program to suit your site's environment and users' needs.

Note: There are a number of references to RACF in the following discussion, which should be taken to mean "RACF or any RACF compatible product".

Please ensure that you review the "Start Options for the Session Controller" and "Session Processing Options" sections which follow for important considerations that will assist you in selecting a suitable processing method.

Security

The Arbutus session controller uses the RACROUTE facility for User ID authentication, and the Sysout API (SAPI) for JES sysout data set retrieval. The

Arbutus software programs must be run from an APF authorized library. As well, all support programs used by the server session must reside in APF authorized libraries. Support programs that may be used are the DB2 and IMS Interface programs and any external sort programs such as Syncsort, DFSort or CA-Sort.

If your system is protected by a firewall, you may need to open a port in the firewall to allow communication between the Arbutus Clients (Analyzer and ConnectPlus) and the Arbutus zSeries Server. Specify this port number in "Step 2 - Edit the ASIINI Data Set" on page 21.

Note: If you will be accessing your Arbutus zSeries Server from external locations you may need to allow appropriate access through your external firewall as well.

OpenEdition Application Services

OpenEdition Application Services running under OS/390 and zSeries provide for the use of sockets. To use OE Application Services you need either:

- An OMVS segment for each TSO user accessing the server, or
- A default OMVS segment for all TSO users. No special privileges, such as superuser status, are required.

Most organizations implementing OE security already have an OMVS segment for TSO users, or a default, and nothing more is required to run the server. If this has not been done at your site, an EC6 abend code may result when the user attempts to connect.

Contact your security administrator for help with adding an OMVS segment. The IBM manual, "OS/390 V2R5.0 OpenEdition Planning," describes the process of creating a default OMVS segment.

Start Options for the Session Controller

When the server is run with a PC-based Arbutus Client, there is a small server program called ASISERV that runs continually on your system as the session controller, listening for connection requests from PC-based Arbutus Clients. This program is run using installed JCL, and can be submitted to your system as required using the SUBMIT command, or can be set up as a started task for automatic submission whenever your zSeries system is started.

Started Task Considerations

Many aspects of running started tasks are a matter of choice based on the standards of any given site, and the discussion included here is for general guidance only; there are no specific requirements for started tasks from the perspective of the session controller.

Two sample started procedures are provided and described in “Step 1a - Edit the Session Controller JCL for Spawned Jobs” on page 18 and in “Step 1b - Edit the Session Controller JCL for Sub-tasking” on page 19.

The security environment of started procedures is defined using a RACF class, not through the USER, PASSWORD, GROUP, and SECLABEL parameters. If these parameters are specified, the started procedure will fail.

To enable started procedures to access the same protected resources that users and groups access, they must have User ID and group identities. By assigning your server started procedure an identity, your installation can grant specific authorization to access-protected resources. As with any other User ID and group name, the User ID and group name that you assign to a started procedure must be defined to RACF using the ADDUSER and ADDGROUP commands.

If a started procedure is executed without associating its name with a RACF-defined User ID and group name, the process runs as an undefined user. The started task can access RACF-protected resources if the universal access authority for the resource is sufficient to allow the requested operation. However, if a started task uses a RACF-protected resource that grants or denies authority based on access list entries, you must associate the started task with a RACF-defined User ID and group name.

For installations that use generic or group profiles to protect resources, establishing an identity for the server can replace the need for individual TSO User IDs and passwords.

Top Secret

A Top Secret facility can be added for Arbutus. This can be done with the following Top Secret commands:

```
TSS MODIFY FAC (USERnn=NAME=facname)
TSS MODIFY FAC (facname=PGM=pgm)
```

where 'nn' is an available Top Secret FACILITY number, 'facname' is the facility name you will be assigning to Arbutus, and 'pgm' is the first three characters of the program name(s) that will be using the PWRESET() function.

For example, you could specify:

```
TSS MODIFY FAC (USER43=NAME=ASI)
TSS MODIFY FAC (ASI=PGM=ASI)
```

This would set up Top Secret facility USER43 with a facility name of ASI and allow program names beginning with ASI to be considered within this facility.

The Arbutus started task userid will then need to have this facility assigned as its master facility as follows:

```
TSS ADD (ASIusrid) MASTFAC (ASI)
```

Userids that will be allowed to reset their passwords through the Arbutus PWRESET() function should have the ASI facility added to their userid or to a common profile shared by all Arbutus users:

```
TSS ADD(userid) FAC(ASI)
```

Starting and Stopping the Session Controller

When running as a started task, the session controller can be started using the START console command:

```
START ASISRV
```

The session controller can be stopped using the STOP or MODIFY console commands.

```
STOP ASISRV
```

```
MODIFY ASISRV, SHUTDOWN
```

where ASISRV is the name of your started task or submitted job. Delays of up to five seconds for processing of the STOP command can occur.

Session Processing Options

When the server is run with a PC-based Arbutus Client, the server session that supports each user connection can be run as either a spawned job, or as a sub-task within the session controller address space.

Selecting the processing option is a very simple, one line configuration setting, so changing the processing option at a later date is not a problem.

There are advantages and disadvantages to both processing methods:

1. Each server session running as a spawned job requires an initiator on your system, just as any submitted job does. If your system is particularly busy, the end-user may see delays in establishing the connection, while waiting for an initiator to come available. If this is the case at your site, selecting sub-tasks as the processing method may improve end-user performance.
2. Use of the optional IMS interface is not supported for server sessions running as sub-tasks. If this interface will be used, the server sessions must be run as spawned jobs.
3. Each server session running as a sub-task is run within the address space of the session controller. Depending on the number of PC-based Arbutus Client users at your site, this can require a very large or unlimited region size associated with your session controller job.

4. Each server session running as a sub-task will use the Owner ID of the session controller as the DB2 authorization ID. If the DB2 interface is used, you must ensure that the Owner ID of the session controller has sufficient authority to access DB2 tables. This restriction applies to DB2 only; access to other resources is authenticated using the User ID of the connecting user.

Licensing and Authorization

Primary licensing of the Arbutus zSeries Server is controlled via the the Licence Agreement contained in the ASILIC dataset. All other authorization and licensing options are controlled by parameters specified in the ASIINI data set. This data set is required for all configurations, and is described in detail in “Step 2 - Edit the ASIINI Data Set” on page 21

Upgrading an Existing System

For versions of the Arbutus zSeries Server that can be upgraded, upgrade JCL named ASIUPGR is included in the SYSFILES PDS found on the Arbutus Website or at Label 1 of the installation tape. The instructions that follow for installing a new system can be used for the upgrade as well.

The Database Administrator must run the BIND process to recreate the DB2 plan for each subsystem to which access is required.

Note: You should rename or back up existing data sets before replacing them.

Installing a New System

If copying Arbutus zSeries Server components from the Arbutus Website, see “Copying Components from the Arbutus Website” on page 12. If copying Arbutus zSeries Server components from tape, see “Copying Components from Tape” on page 14.

Copying Components from the Arbutus Website

This section describes the process for copying the Arbutus zSeries Server components from the Arbutus Website.

Step 1 - Download file from the Arbutus Website

All of the components have been placed into a zipped folder on the Arbutus Website:

<http://www.arbutussoftware.com/support/downloads/>

The download file is named:

`Mainframe_Server.zip`

Once downloaded, the folder contents must be unzipped before transferring them to your mainframe system.

The following procedures use the TSO RECEIVE command to repackage a data set into its original format. Instructions for the TSO RECEIVE command are described later in this document.

Note: All sizes in the tables below refer to a 3390 DASD device.

Step 2 - Transfer all necessary data sets to the mainframe

Please perform a binary transfer of all of the required data sets and any of the necessary optional data sets shown in the tables below to your mainframe system.

Note: Change prefix1 in the final data set names in the tables below to be the prefix selected for read-only server system data sets. The second and third-level-qualifier names that appear in final data set names are the default values. These names should not be changed, if possible, because the default names are used descriptively throughout Arbutus documentation.

The data sets listed in Table 1 below are required for all installations.

Table 1: Data sets required for the Arbutus zSeries Server:

Unzipped File Name	Final Data Set Name	Desc	Size	DCB Attributes	Method
SHIPSYSF	prefix1.ASI.SYSFILES	JCL to install and run the system	7 tracks	FB, 80, PO	Receive
SHIPLOAD	prefix1.ASI.LOADLIB	Executable programs	3 cylinders	U, 0, PO	Receive
SHIPLIC	prefix1.ASI.LICENCE	Licence Agreement	1 track	VB, 32756, 32760	Transfer (Binary)

The data set listed in Table 2 is only required if you will be using the DB2 Interface.

Table 2: Optional data set required for the DB2 Interface:

Unzipped File Name	Final Data Set Name	Desc	Size	DCB Attributes	Method
SHIPDBRM	prefix1.ASI.DBRMLIB	DB2 Data Base Resource Module	1 track	FB, 80, PO	Receive

Tip: If you are upgrading an existing version of the Arbutus zSeries Server we recommend that you either make a copy of your existing version, or RECEIVE the new data sets into a new version specific set of data sets.

Step 3 - Rebuild necessary data sets on the mainframe

When the data sets have been uploaded to the mainframe, please run the TSO RECEIVE process for those uploaded data sets that need to be rebuilt (i.e., those with "Receive" in the Method columns shown in the tables above).

The TSO RECEIVE will build the output data set with DCB attributes identical to the DCB attributes of the original data set.

```
==> RECEIVE inds ('your.uploaded.file')
```

in response to the "Enter restore parameters" prompt, enter

```
DSN ('xxxxxxxx')
```

Where "xxxxxxxx" is the fully qualified data set name that will be created by the TSO RECEIVE process.

Tip: We recommend that you use the final data set names listed in the tables above for your output data set names, as these are the names used throughout this Installation Guide.

After completing the preceding instructions for transferring the Arbutus zSeries Server components from the Arbutus Website to your mainframe system, see "Required Installation Procedures" on page 17 to begin the installation of the Arbutus zSeries Server.

Copying Components from Tape

This section describes the process for copying the Arbutus zSeries Server components from an installation tape.

Installation Tape Contents

If copying the Arbutus zSeries Server components from tape, the installation tape supplied is a standard 3490 tape cartridge containing the following data sets in Label sequence:

Label	DSN	Content
1	ASI.SYS.SYSFILES	System files
2	ASI.SYS.LOADLIB	Load library with program modules
3	ASI.SYS.DBRMLIB	DB2 Plan ASIDB2
4	ASI.SYS.LICENCE	Licence Agreement

Sites using standard security software, such as RACF or TSS, will require READ-ONLY access to all data sets.

Step 1 - Create JCL to Copy the Install Procedure from Tape

Create a batch job to copy the SYSFILES PDS from Label 1 of the installation tape. The SYSFILES PDS is a permanent component of the Arbutus Server, so the high level qualifier used here should be the same as will be used for the remainder of the system data sets. A sample jobstream is shown below:

```

//Jobname JOB (Accounting Information),'COPYASI',           Review 1
//  TIME=(3,0)
//STEP01  EXEC PGM=IEBCOPY
//SYSUT1  DD DSN=ASI.SYS.SYSFILES,
//          DISP=OLD,
//UNIT=3490,                                           Review 2
//          VOL=SER=volser,                             Review 3
//          LABEL=(1,SL)
//SYSUT2  DD DSN=prefix1.ASI.SYSFILES,                 Review 4
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,                                 Review 5
//          DCB=(RECFM=FB,BLKSIZE=27920,LRECL=80),
//          SPACE=(TRKS,(15,2,10),RLSE),EXPDT=EXPIRY   Review 6
//SYSIN   DD *
//          COPY INDD=((SYSUT1,R)),OUTDD=SYSUT2,LIST=NO
//SYSUT3  DD UNIT=WORK,SPACE=(CYL,(1,1))
//SYSUT4  DD UNIT=WORK,SPACE=(CYL,(1,1))
//SYSPRINT DD SYSOUT=*
//

```

Modify the items marked Review as follows:

Review 1

Change the Jobname and Accounting Information to reflect the standards in effect at your site. Optional job card parameters may be required to specify or override default values.

Review 2

Change 3490 to specify the UNIT designation for tape cartridge at your site.

Review 3

Change volser to reflect the appropriate volume serial number shown on the installation tape cartridge.

Review 4

Change prefix1.ASI.SYSFILES to specify the output data set where the SYSFILES PDS from Label 1 will be written.

Review 5

Change SYSDA to specify the UNIT designation for permanent disk.

Review 6

If required at your site, change the expiry parameter to specify an expiry date for your output data set.

If no expiry date is required, remove the expiry parameter and the comma following the space parameter.

Step 2 - Submit the Copy JCL

Submit the copy job created in Step 1, and verify the SYSFILES PDS has been successfully copied from Label 1 of the installation tape to the file name specified at DD SYSUT2.

Step 3 - Edit the Install Procedure

Edit the member ASIINST for new installations. Edit the member ASIUPGR for upgrading an existing system.

Note: The ASIUPGR JCL is only included in the installation package for versions that can be upgraded. If a full install is required, this JCL has been omitted.

Modify the items described below to reflect naming conventions and standards in effect at your site. These changes can be accomplished using the global change facility available under TSO.

The syntax for the TSO change command is:

```
CHANGE 'XXXXX' 'yyy' ALL
```

where XXXXX is the old value, and yyy is the new value, and the delimiter between command words is a blank space.

1. Modify the Jobname and Accounting information.
2. Establish data set names for all files to be copied.
 - The first-level qualifier can be globally modified by changing prefix1 to the prefix selected for read-only server system data sets.

- The second and third-level-qualifier names that appear in the install job are the default values. These names should not be changed, if possible, because the default names are used descriptively throughout Arbutus documentation.
3. Change 3490 to specify the UNIT designation for tape cartridge at your site.
 4. Change SYSDA to specify the UNIT designation for permanent disk.

Step 4 - Run the Install Procedure

Submit the install job modified in Step 3, and verify the successful completion of all steps.

If the condition code from any step is greater than zero, execution is halted. Possible reasons for non-zero condition codes are space allocation problems or an incorrect JCL specification.

After the problem has been corrected, the job may be restarted, or all data sets cataloged by the prior execution of this job must be deleted, and the job rerun from the beginning.

The following server data sets are created by the install procedure:

Default Name	Description
prefix1.ASI.LOADLIB	ASI program modules
prefix1.ASI.DBRMLIB	Data Base Resource Module ASIDB2
prefix1.ASI.LICENCE	Licence Agreement

Note: Except for the changes described later in this guide, do not modify the server data sets in any way.

After completing the preceding instructions for transferring the Arbutus zSeries Server components from tape to your mainframe system, see “Required Installation Procedures” on page 17 to begin the installation of the Arbutus zSeries Server.

Required Installation Procedures

Once the Arbutus zSeries Server components have been successfully placed on the mainframe, the following steps are required to install or upgrade the server.

Note: If running server sessions as sub-tasks, please proceed directly to “Step 1b - Edit the Session Controller JCL for Sub-tasking” on page 19.

Step 1a - Edit the Session Controller JCL for Spawned Jobs

The session controller JCL to run user connections as spawned jobs is included in the installed System File PDS, member ASISERV. A started task to run user connections as spawned jobs is included in the installed System File PDS, member ASISTR.

```
//ASISERV JOB ( ), 'ASISERV',MSGCLASS=D,CLASS=A,
//  USER=USER,PASSWORD=PASSWORD                               Review 1
//STEP01  EXEC PGM=ASISERV,DYNAMNBR=35,REGION=4096K
//*-----
//* ARBUTUS SESSION CONTROLLER - SPAWNED JOBS
//*
//* NOTE: ALL STEPLIB LIBRARIES MUST BE APF AUTHORIZED
//*-----
//STEPLIB  DD DSN=prefix1.ASI.LOADLIB,DISP=SHR                 Review 2
//ASIINI  DD DSN=prefix1.ASI.SYSFILES(ASIINI),DISP=SHR        Review 3
//ASILIC  DD DSN=prefix1.ASI.LICENSE,DISP=SHR                 Review 4
//ASIJCL  DD DSN=prefix1.ASI.SYSFILES(ASISESS),DISP=SHR       Review 5
//IMSJCL  DD DSN=prefix1.ASI.SYSFILES(IMSSESS),DISP=SHR      Review 6
//SYSTEM  DD SYSOUT=*                                         Review 7
//SYSPRINT DD SYSOUT=*
//STDERR  DD SYSOUT=*
//TCPLOG  DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSUT2  DD SYSOUT=(*,INTRDR),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//
```

Modify the items marked Review above to reflect the naming conventions, standards and processing options for your site:

Review 1

For submitted JCL, specify the following information:

- a job name and appropriate accounting information
- an appropriate MSGCLASS parameter
- an appropriate CLASS parameter for continuously running tasks
- a User ID to be the owner of the sessions
- a password for the User ID

Review 2

Specify the Arbutus load library installed at your site.

When running spawned jobs, all libraries in the STEPLIB concatenation must be APF authorized, as this is required for the Sysout API (SAPI) used to retrieve JES information.

Review 3

Specify the Arbutus ASIINI Data Set installed at your site.

Review 4

Specify the Arbutus ASILIC Data Set installed at your site. This file must not be altered.

Note: The first time an Arbutus client is used to connect to the Arbutus zSeries Server, the user will need to complete and accept the Arbutus license agreement using the serial number contained in the provided ASIINI file. As a result, the user must have update authority to the ASILIC data set to complete the license agreement process.

Review 5

Specify the installed SYSFILES PDS, member ASISESS.

Review 6

Specify the installed SYSFILES PDS, member IMSSESS. If the IMS Interface will not be used, this DD statement can be removed.

Review 7

DD cards SYSTERM, SYSPRINT, STDERR, TCPLOG, and SYSTSPRT will contain various server and system messages. The TCPLOG is where log messages are written.

Note: When all review items above have been modified as necessary, please proceed to "Step 2 - Edit the ASIINI Data Set" on page 21.

Step 1b - Edit the Session Controller JCL for Sub-tasking

The session controller JCL to run server sessions as sub-tasks is included in the installed System File PDS, member ASISERVS.

A started task to run user connections as sub-tasks is included in the installed System File PDS, member ASISTRTS.

```
//ASISERVS JOB (), 'ASISERVS',MSGCLASS=D,CLASS=A                               Review 1
//STEP01 EXEC PGM=ASISERV,DYNAMNBR=35,REGION=0M,TIME=NOLIMIT
//*-----
//* ARBUTUS SESSION CONTROLLER - SUBTASKS
//*
//* NOTE: ALL STEPLIB LIBRARIES MUST BE APF AUTHORIZED
//*-----
//STEPLIB DD DSN=prefix1.ASI.LOADLIB,DISP=SHR                                Review 2
//* DD DSN=DSN610.SDSNLOAD,DISP=SHR
//* DD DSN=IMS.RESLIB,DISP=SHR
//ASIINI DD DSN=prefix1.ASI.SYSFILES(ASIINI),DISP=SHR                       Review 3
//ASILIC DD DSN=prefix1.ASI.LICENSE,DISP=SHR                                Review 4
//*-----
//* FOR IMS - ALL LIBRARIES IN IMS CONCATENATION MUST BE APF AUTHORIZED
//*-----
//*IMS DD DSN=IMS.PSBLIB,DISP=SHR                                           Review 5
//* DD DSN=IMS.DBDLIB,DISP=SHR
//*ASIDBD DD DSN=IMS.DBDSRCE,DISP=SHR                                       Review 6
//*DFSVSAMP DD DSN=IMS.PROCLIB(DFSVSAMP),DISP=SHR                           Review 7
//*DFSRESLB DD DSN=IMS.RESLIB,DISP=SHR
```

```

//*-----
//ASISRV01 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE)
//STDERR DD SYSOUT=*
//TCPLOG DD SYSOUT=* Review 8
//*-----
//* FOR META DATA CREATION - COBOL, PL/1, and EASYTRIEVE
//*-----
//*ASICBL DD DSN=PRD.COBO.LIB,DISP=SHR Review 9
//*ASIPL1 DD DSN=PRD.PL1.LIB,DISP=SHR
//*ASIEZT DD DSN=PRD.PL1.LIB,DISP=SHR
//

```

Modify the items marked Review above to reflect the naming conventions, standards and processing options for your site:

Review 1

For submitted JCL, specify the following information:

- a job name and appropriate accounting information.
- an appropriate MSGCLASS parameter.
- an appropriate CLASS parameter for continuously running tasks.

Review 2

Specify the Arbutus load library installed at your site. If the DB2 or IMS interfaces will be used, and these programs are not in a globally accessible location such as the LPA, these must be included in the STEPLIB concatenation.

Note: The DB2 and IMS load library names in the JCL illustration are examples only; please replace these names with the appropriate names at your site.

All libraries in the STEPLIB concatenation must be APF authorized, as this is required for RACROUTE User ID and password verification at the time of connection.

Review 3

Specify the Arbutus ASIINI Data Set installed at your site.

Review 4

Specify the Arbutus ASILIC Data Set installed at your site. This file must not be altered.

Note: The first time an Arbutus client is used to connect to the Arbutus zSeries Server, the user will need to complete and accept the Arbutus license agreement using the serial number contained in the provided ASIINI file. As a result, the user must have update authority to the ASILIC data set to complete the license agreement process.

Review 5

Specify the PSB and DBD libraries that contain the PSB named in the DFSRRCOO parameters, and the data base definitions referenced by the PSB.

Review 6

Specify the PDS that contains the DBD source for data bases to be accessed. Only the DBD source for data bases that will be accessed by the end-user need be included in this PDS.

Review 7

Specify the data set containing the IMS control statements for buffer allocation.

Review 8

DD cards STDERR and TCPLOG will contain various server and system messages. In particular, the TCPLOG is where log messages are written.

Note: When all review items above have been modified as necessary, please proceed to "Step 2 - Edit the ASIINI Data Set" on page 21.

Optional JCL for Sub-Tasking:

```

/** Optional Libraries                                     Review 9
//ASICBL DD DSN=PRD.COBO.LIB,DISP=SHR
//ASIPL1 DD DSN=PRD.PL1.LIB,DISP=SHR
//ASIEZT DD DSN=PRD.PL1.LIB,DISP=SHR
//

```

Review 9

The ASICBL, ASIPL1 and ASIEZT DD statements are used to allocate COBOL, PL/1 and EASYTRIEVE libraries that contain members to be converted to table layouts using Analyzer.

Step 2 - Edit the ASIINI Data Set

The ASIINI Data Set is required for all configurations of the server. The ASIINI Data Set is installed in 'prefix1.ASI.SYSFILES', member ASIINI. The parameters in this data set control a variety of performance, licensing and end-user preferences that allow you to customize the server software to suit your environment.

A typical example of an ASIINI Data Set is shown below:

```

[Arbutus Stub Info]
PORT=
TIMEOUT=15
LOG=1
ENCRYPT
JPREFIX=ASI

[Arbutus Server Info]
INITIATORWAIT=1
USERS=
DESIGNATION=Arbutus Software Inc.
SERIALNO=

```

```
EXPIRY=  
EXPCODE=  
SRVCODE=
```

Licensing parameters in the ASIINI Data Set must be updated according to the licensing information e-mail that accompanies this installation guide.

The ASIINI Data Set is read each time a session is started. Changes made while the session is running are not recognized until a new session is started.

The following is a detailed description of each of the required ASIINI Data Set parameters.

Note: The case for each of the ASIINI data set parameter values must be as shown. An ASIINI data set parameter will not be recognized if the case is changed.

Required ASIINI Parameters [Arbutus Stub Info]

Port Number

Specify the TCP/IP port number on which the session controller should listen for connection requests. The default syntax is:

```
PORT=
```

Any available port number in the range 1024 through 32767 may be specified. The value entered here must match the port number entered for the server profile created in the Analyzer client.

Maximum Inactivity Wait Time

Specify the amount of time, in minutes, that you want the session to wait for activity from a PC-based Arbutus Client. The default syntax is:

```
TIMEOUT=15
```

The default value for this setting terminates the sessions after 15 minutes of inactivity. The minimum value for this setting is 5 minutes. Anything less than that may cause the session to end during the course of normal processing. To achieve no timeout, set the value to 999.

This setting applies only to the maximum length of time that the server waits for input from the client, and has no effect on times associated with long-running processes.

Note: This value may be reduced by the client in the server profile, as required. For more information on server profiles see the Analyzer Installation Guide and the Analyzer Online Help.

Message Logging

Specify the level of message logging to be used by the session controller. The default syntax is:

```
LOG=1
```

A value of 1 specifies logging all connection attempts, a value of 2 specifies full message logging. Full message logging includes a great deal of information, and should be used only for troubleshooting purposes. A value of 0 indicates no message logging.

Encryption

Specifies that all data transferred between servers or from server to local must be encrypted.

The default syntax is:

```
ENCRYPT=YES
```

The presence of the keyword Encrypt ensures that all data transferred between servers or from server to local must be encrypted.

Note: If encryption is mandated in the server INI then the server profile used to access the server must also have encryption enabled or the end user will not be allowed to connect to the server and will be warned to enable encryption in their server profile.

To turn off mandatory encryption, the keyword Encrypt must be removed from the ASIINI Data Set.

JOBNAME Prefix

This value indicates whether server sessions are to be processed as sub-tasks or spawned jobs. It also specifies the jobname that spawned jobs are run under. The syntax is:

```
JPREFIX=
```

To indicate that server sessions are to be run as sub-tasks, specify either all blanks or NOPASSCHECK for this value:

- when blanks are specified, sub-tasks will be run under the security environment established based on the User ID of the connecting client.
- when NOPASSCHECK is specified, sub-tasks will be run under the security environment of the session controller. In this case, the User ID and password sent from the connecting client will not be verified.

To indicate that server sessions are to be run as spawned jobs, specify a jobname prefix according to the following guidelines:

- the maximum length of the session jobname prefix is 7 characters, and shorter jobname prefixes are recommended to allow a larger range of unique jobnames to be generated.
- the jobname prefix can be specified as &SYSUID, in which case the jobname prefix will be set equal to the User ID of the connecting user.
- by default, a number in the range 1 through 999 will be added to the jobname prefix to create a unique job name.

- to qualify the job session name with a letter instead of a number include the JAlpha=1 parameter in your INI file. In this case, the values "A" through "ZZ" will be added to the jobname prefix, for a maximum of 52 unique concurrent sessions.

Required ASIINI Parameters [Arbutus Server Info]

Designation

Contains the title to appear at the top of reports generated when running with the offline interface. This entry can be changed to any value or left blank. Users can temporarily change the value using the server command SET DESIGNATION. The default syntax is:

```
DESIGNATION=Arbutus Software Inc.
```

Serial Number

Indicates the copy number of your Arbutus Analyzer software. The syntax is:

```
SERIALNO=
```

This value must be entered exactly as provided in your licensing information e-mail that accompanies this installation guide.

Expiry Code

Indicates the expiry date of your server software. The syntax is:

```
EXPCODE=
```

This value must be entered exactly as provided in your licensing information e-mail that accompanies this installation guide.

Expiry

Shows a printable version of your software expiry date. The syntax is:

```
EXPIRY=
```

CPU ID Code

Indicates the CPU serial number for your system. The syntax is:

```
SRVCODE=
```

This value must be entered exactly as provided in your licensing information e-mail that accompanies this installation guide. If you are licensing more than one CPU, this parameter should be omitted and replaced with:

```
SRVCODE1=
```

```
SRVCODE2=
```

to a maximum of 20 unique CPU license codes.

Authorized Seat Count

This value should be updated with the “Authorized Seat Count” value specified in the licensing information e-mail. The syntax is:

USERS=

Start Wait Time

Specify the amount of time, in minutes, that the server should wait for the spawned job to start before it will cancel the job. This parameter is ignored when running server sessions as sub-tasks. The default syntax is:

INITIATORWAIT=1

The session job may be delayed because the job is waiting for an initiator (the specific class of initiators to use for the session job is indicated on the CLASS parameter of the job card).

The value set should be long enough to account for any reasonable delay, but not so long that the job is left pending for an extended period of time.

Optional ASIINI Parameters

The following INI parameters are not required, but may be included to override the default values. Unless otherwise specified, all of these INI parameters must be entered in the [Arbutus Server Info] section of the INI file.

DB2 Authorization

This parameter is required to use the optional DB2 data interface. The syntax is:

DB2AUTH=nnnnnnnnnnnn

The 12 digit license code (nnnnnnnnnnnn) must be entered exactly as shown in your licensing e-mail that accompanies this installation package. For additional details on DB2 INI parameters see “The DB2 Interface” on page 35.

IMS Authorization

This parameter is required to use the optional IMS data interface. The syntax is:

IMSAUTH=nnnnnnnnnnnn

The 12 digit license code (nnnnnnnnnnnn) must be entered exactly as shown in your licensing e-mail that accompanies this installation package. For additional details on IMS INI parameters see “The IMS Interface” on page 40.

JES Listings

User sessions run as spawned jobs are automatically deleted from the system when the session ends. Specify this parameter to retain the JES listings for completed jobs. The default syntax is:

JESDELETE=NO

This parameter must be entered in the [Arbutus Stub Info] section.

Memory Size

Use this parameter to change the default address space size (2,000,000) for user sessions. The syntax is:

```
MEMSIZE=nnnnnnn
```

The minimum size that can be specified is 50000 and the maximum that can be specified depends on the amount of virtual storage available.

Buffer Size

Use this parameter to change the default data buffer size (40,000) for user sessions. The syntax is:

```
BUFSIZE=nnnnn
```

The minimum data buffer size that can be specified is 32760, and should never be more than one third of the Memsize value.

Binary Allocation

Use this parameter to change the default values for data file allocation. The syntax is:

```
BINALLOC=blksize,primary-space,secondary-space,primary-unit,work-  
unit,allocation-unit,↓
```

Parameters are positional, and so must be specified in the order shown, separated by a single comma. If not all parameters are required, use an asterisk as a placeholder.

The SET ALLOCATION command can be used to override the default values for a specific end user process.

The blksize default value is 27998. The maximum that can be specified is 32760, and the minimum is 1.

The primary-space default value is 100. The primary space allocation unit is TRACKS unless overridden as described below.

The secondary-space default value is 1000. The secondary space allocation unit is TRACKS unless overridden as described below.

The default unit value for permanent data files (primary-unit) is selected based on defaults associated with the current User ID. Only the DASD unit can be specified here; the SET VOLUME command is available to indicate a specific DASD volume or group of volumes.

The default unit value for work data files (work-unit) is selected based on defaults associated with the current User ID. Only the DASD unit can be specified here; the SET VOLUME command is available to indicate a specific DASD volume or group of volumes.

The allocation unit (allocation-unit) default value is TRACKS. Supported allocation units are TRK, CYL or BLOCK.

Text Allocation

Use this parameter to change the default values for text file allocation. The syntax is:

```
TX TALLOC=blksize,primary-space,secondary-space,primary-unit,allocation-
unit,lrecl
```

Parameters are positional, and so must be specified in the order shown, separated by a single comma. If not all parameters are required, use an asterisk as a placeholder.

The blksize default value is 27998. The maximum that can be specified is 32760, and the minimum is 1.

The primary-space default value is 100. The primary space allocation unit for text files is BLOCKS unless overridden as described below.

The secondary-space default value is 500. The secondary space allocation unit for text files is BLOCKS unless overridden as described below.

The default unit value for permanent text files (primary-unit) is selected based on defaults associated with the current User ID.

The allocation unit (allocation-unit) default value is BLOCKS. Supported allocation units are TRK, CYL or BLOCK.

The default logical record length (lrecl) for text files is 255.

Personal File Qualifier

Use this entry to specify a high-level qualifier for personal files at sites that do not allow personal data sets. The syntax is:

```
PPREFIX=
```

If this entry is omitted, personal data sets will be created using the USER ID of the connecting user as the high-level qualifier. Examples of personal data sets created by Arbutus Analyzer are the SORT work file and intermediate INDEX files.

System Sort Program

Use this parameter to change the default external sort program name of SORT. The syntax is:

```
SORTPGM=
```

The name must be entered in upper case. The specified sort load module must be in a library that is accessible to the Arbutus product. For additional details see "System Sort" on page 43.

System Sort Parameters

Use this parameter to change the default sort parameters of "OPTION MSGPRT=CRITICAL". The syntax is:

```
SORTPARAM=
```

For additional details see "System Sort" on page 43.

System Sort DD Names

Use this parameter to change the Sort DD names from the Syncsort default values of \$ORTPARM,SORTIN,SORTOUT,SYSin,SYsOUT. The syntax is:

```
SORTDD=
```

PL/1 Margins

Use this parameter to override the default PL/1 margin specification of FMARGINS (fixed margins). The default syntax is:

```
PL1=VMARGINS
```

Step 3 - Edit the Session JCL

This step can be skipped if you are planning to process client connections as sub-tasks.

The session JCL is included in the System File PDS, member ASISESS. This JCL is dynamically configured and submitted via the internal reader when a connection request is received by the server program.

The %S values in the JCL represent items that are modified at the time the job is submitted; do not remove these or modify the sequence of the job card parameters. Additional job card parameters can be added at the end of either of the job card lines, and additional job card lines can be added to the session JCL.

```
//%S JOB (%S), 'ASISESS', MSGLEVEL=(0,0), CLASS=A, TIME=NOLIMIT,      Review 1
//      MSGCLASS=D, USER=%S, PASSWORD=%S
//*****
//* MODEL JCL FOR SPAWNED SERVER SESSIONS
//*****
//STEP01 EXEC PGM=ASIMAIN, DYNAMNBR=35, REGION=8192K, TIME=(10)
//STEPLIB DD DSN=prefix1.ASI.LOADLIB, DISP=SHR                      Review 2
//*
//*-----
//* ADD THE FOLLOWING IF YOUR TCPIP ADDRESS SPACE IS NOT CALLED TCPIP
//* SYSTCPD DD DSN=ASI.TCPIP.DATA, DISP=SHR
//* THIS DATA SET IS A COPY OF YOUR TCPIP CONFIGURATION DATA SET
//* CONTAINING THE SPECIFICATION OF THE TCPIP ADDRESS SPACE NAME:
//* TCPIPJOBNAME XXXX
//*-----
//ASIINI DD DSN=prefix1.ASI.SYSFILES (ASIINI), DISP=SHR              Review 3
//SYSTSPRT DD SYSOUT=*
//SYSTEMM DD SYSOUT=*
//TCPLOG DD SYSOUT=*
```

```
//ASISRV01 DD *
/*
/*EOF
```

Modify the items marked Review to reflect naming conventions and standards in effect at your site:

Review 1

Change the accounting information ('Account Info') to specify valid accounting information at your site, and modify the CLASS=A setting, if required, to specify a valid class for initiators on your system.

Modify the MSGCLASS and MSGLEVEL values to reflect values appropriate for your site. These specify the output destination of the JES listing and the level of JES messages recorded.

Review 2

Specify the Arbutus load library installed at your site. If the DB2 interface will be used and the DB2 routines are not in a globally accessible location such as the LPA, the DB2 load library should be included in the STEPLIB concatenation.

Review 3

Specify the Arbutus ASIINI Data Set installed at your site.

Optional Session JCL:

```
/** Optional Libraries Review 4
//ASICBL DD DSN=PRD.COBOBOL.LIB,DISP=SHR
//ASIPL1 DD DSN=PRD.PL1.LIB,DISP=SHR
//ASIEZT DD DSN=PRD.PL1.LIB,DISP=SHR
//
```

Review 4

The ASICBL, ASIPL1 and ASIEZT DD statements shown above are used to allocate COBOL, PL/1 or EASYTRIEVE libraries that contain members to be converted to table layouts using Analyzer.

Optional Data Sets

Other optional DD statements may need to be added to your session JCL, depending on the configuration options selected. If your ASIINI Data Set specifies DD names instead of data set names, the DD statements and associated data sets allocations must also be included.

Optional Installation Procedures

Installation steps 4 through 7 shown below, describe set-up requirements for features that may or may not be used at your site.

Steps 4 through 7 pertain to the DB2, IMS and Adabas interfaces, and can be ignored if these will not be used.

Step 8 is required if your data files are defined using COBOL, PL/1 or EASYTRIEVE and you would like to automatically create the Arbutus table layouts using these definitions.

Note: None of these optional steps are required for successful operation of the server, and can be performed later.

Step 4 - Edit the IMS Session JCL

This step can be skipped if the IMS interface is not enabled.

The IMS session JCL is included in the System File PDS, member IMSSESS. This JCL is dynamically configured and submitted when a connection request with the IMS access flag enabled is received by the server program.

The %S values in the JCL represent items that are modified by the server program at the time the job is submitted; do not change these or modify the sequence of the job card parameters. Additional job card parameters can be added at the end of either line, or additional job card lines can be added.

```

//%S JOB ('Account Info'), 'IMSSESS', MSGLEVEL=(0,0), CLASS=A, TIME=NOLIMIT,
//      MSGCLASS=D, USER=%S, PASSWORD=%S                                Review 1
//*****
//* MODEL JCL FOR SPAWNED IMS SERVER SESSIONS
//*****
//STEP01 EXEC PGM=DFSRRRC00, DYNAMNBR=35, REGION=8192K, TIME=(40,0),
//      PARM='DLI,ASIMAIN,ASIPSB01'                                    Review 2
//STEPLIB DD DSN=IMS.RESLIB, DISP=SHR                                Review 3
//      DD DSN=prefix1.ASI.LOADLIB, DISP=SHR                        Review 4
//*-----
//* IMS DATA SETS
//*-----
//DFSRESLB DD DSN=IMS.RESLIB, DISP=SHR
//IMS      DD DSN=IMS.PSBLIB, DISP=SHR                                Review 5
//      DD DSN=IMS.DBDLIB, DISP=SHR
//IEFRDER DD DUMMY
//SYSUDUMP DD DUMMY
//IMSLOGR DD DUMMY
//IMSMON  DD DUMMY
//DFSVSAMP DD DSN=IMS.PROCLIB(DFSVM02), DISP=SHR                    Review 6
//ASIDBD  DD DSN=IMS.DBDSRCE.LIB, DISP=SHR                          Review 7
//*-----
//* ASI DATA SETS
//*-----
//ASIINI  DD DSN=prefix1.ASI.SYSFILES(ASIINI), DISP=SHR              Review 8
//SYSPRT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//TCPLOG  DD SYSOUT=*
//ASISRV01 DD *
/*EOF

```

Modify the items marked Review as follows to reflect naming conventions and standards in effect at your site:

Note: Additional information regarding the IMS setup requirements are described in “The IMS Interface” on page 47.

Review 1

Modify the MSGCLASS and MSGLEVEL values. These specify the output destination of the JES listing and the level of JES messages recorded.

Review 2

Replace ASIPSB01 with the name of the IMS PSB created for use by the session.

Note: The IMS session can be run as either a DLI or BMP region.

Review 3

Specify the library from which the IMS modules are to be loaded.

Review 4

Specify the Arbutus load library installed at your site.

Review 5

Specify the PSB and DBD libraries that contain the PSB named in the DFSRRC00 parameters, and the data base definitions referenced by the PSB.

Review 6

Specify the data set containing the IMS control statements for buffer allocation.

Review 7

Specify the PDS that contains the source for your IMS data bases. Only the DBD source for data bases that will be accessed by the end-user need be included in this PDS.

Review 8

Specify the ASIINI Data Set installed at your site.

Optional IMS Session JCL:

```

/** Optional Libraries Review 9
//ASICBL DD DSN=PRD.COBO.LIB,DISP=SHR
//ASIPL1 DD DSN=PRD.PL1.LIB,DISP=SHR
//ASIEZT DD DSN=PRD.PL1.LIB,DISP=SHR
//

```

Review 9

The ASICBL, ASIPL1 and ASIEZT DD statements are used to allocate COBOL, PL/1 and EASYTRIEVE libraries that contain members to be converted to table layouts using Analyzer.

Optional Data Sets

Other optional DD statements may need to be added to your session JCL, depending on the configuration options selected. If your ASIINI Data Set specifies DD names instead of data set names, the DD statements and associated data sets allocations must also be included.

Step 5 - Establish Default DB2 Values

You can omit this step if the DB2 interface is not required.

See “The DB2 Interface” on page 35.

Step 6 - Establish the Default IMS Values

You can omit this step if the IMS interface is not required.

See “The IMS Interface” on page 40.

Step 7 - Establish Default Adabas Values

You can omit this step if the ADBAS interface is not required.

See “The Adabas Interface” on page 42.

Step 8 - Allocate COBOL, PL/1 and EASYTRIEVE DBD Source Libraries

Analyzer can automatically convert COBOL, PL/1 and EASYTRIEVE file layouts to Analyzer table layouts. To simplify this process for the end-user, the libraries containing these file layouts can be allocated to the ASICBL, ASIPL1 and ASIEZT DD names. For more information:

- see Review Note 9 in “Step 1b - Edit the Session Controller JCL for Sub-tasking” on page 19
- see Review Note 4 in “Step 3 - Edit the Session JCL” on page 28 “Installing the zSeries Server” on page 8
- see Review Note 9 in “Step 4 - Edit the IMS Session JCL” on page 30

Start the Session Controller Job

This job will run continuously, listening for connection requests from a PC-based Arbutus Client.

Review the TCPLOG written by the session controller, to ensure that your server is running and the configuration is correct. A typical TCPLOG will look something like:

Mar 28, 2004 09:43:43:

- the port number is incorrect

Other messages may appear and are self-explanatory:

```
05/09/05 16:04:20 *** ERROR: Invalid password.
```

```
05/09/05 16:05:25 *** ERROR: User ARBHH7 is not defined.
```

```
05/09/05 16:05:25 *** ERROR: Expired password.
```

```
05/09/05 16:07:49 *** ERROR: JCL error for job JOB07222.
```

```
05/09/05 16:07:49 IEF212I ASI43 STEP01 STEPLIB - DATA SET NOT FOUND.↓
```

When a JCL error is detected, additional information from the JES sysout data sets is written to the session controller TCPLOG, to assist in diagnosing the problem.

CHAPTER 3 OPTIONAL ZSERIES INTERFACES

The DB2 Interface

The DB2 interface is enabled by modifying values in the ASIINI Data Set:

```
'prefix1.ASI.SYSFILES(ASIINI)'
```

Note: The DB2 Authorization Code provided in the Licensing e-mail accompanying this guide must be entered in the ASIINI Data Set. Any modifications to this value will disable the DB2 interface. For more information see "DB2 Authorization" on page 25.

The DB2 interface uses the standard DSNALI, DSNHLI2, and DSNTIAR interface routines which the Arbutus zSeries Server loads and unloads as required. These modules must reside in a library that the Arbutus zSeries Server can access.

Using DB2 Version 9.1 ODBC

If the DB2 Version 9.1 for z/OS ODBC driver is installed at your site, you can choose to use that instead of the DB2 Call Attach Facility to interface with your DB2 Database.

Note: This may provide a slight performance advantage and will enable the use of the "Edit SQL Select" and "Validate" Where and Order clause features when defining DB2 tables using Analyzer's Data Definition Wizard.

To enable use of ODBC, add ODBC at the end of your DB2Auth entry in the ASIINI file, as shown below:

```
DB2AUTH=123456789012,DB2PLN,ODBC
```

For more information see "Additional Options for ODBC" on page 36.

Creating the DB2 Plan

The Database Administrator at your site must grant Read Only access to any DB2 tables or views required, and must run the BIND process to create a DB2 application plan for each subsystem for which access is required.

The DB2Auth= parameter in the ASIINI file must be updated to include the DB2 Plan name. For example:

```
DB2AUTH=123456789012,DB2PLN
```

The DB2 Plan name is separated from the DB2 authorization code by a single comma, and must be entered in upper case.

The application plan for the Arbutus zSeries Server is created using member ASIDB2 from the installed DBRM library ('prefix.ASI.DBRMLIB').

When using the CAF Interface (that is, when NOT using the ODBC interface), the plan should be bound with the following two bind options:

- VALIDATE(RUN) to indicate that authorization will be checked at run-time

- ISOLATION(CS) to indicate that database values read by Arbutus zSeries Server are protected only while being used.

When binding the plan for use with the ODBC interface, there are a number of packages that must be included. Sample BIND JCL is provided in the SYSFILES PDS, member DB2BIND9.

Specifying DB2 Subsystems

Modify the DB2Systems= parameter in the ASIINI Data Set to specify the DB2 subsystems at your site. These values provide the location for tables and views when accessing DB2 data.

The subsystems will be displayed in the order they are entered in the ASIINI Data Set, so place the most frequently accessed subsystem(s) at the beginning of the list. Enter subsystem names in upper case, with a maximum of 4 characters each; names should be separated by one comma.

Additional Options for ODBC

When using the ODBC interface (via the DB2 Version 9.1 for z/OS ODBC driver), the DB2 ODBC initialization file is available to override various system default values. This data set should be created as fixed length, 72 byte records, and allocated to DD DSNAOINI in your session JCL or started task server JCL. A sample DSNAOINI file is included in the SYSFILES PDS, member DSNAOINI and is shown below:

```

; MUST NOT contain sequence numbers.
; Square brackets are x'ad' x'bd'.
; ***** Common section
[COMMON]
MVSDEFAULTSSID=DB9G                                Review 1
CONNECTTYPE=1                                       Review 2
; ***** Subsystem Section
[DB9G]                                               Review 3
PLANNAME=ASIODBC                                    Review 4
; ***** Data Source Section
[DALLAS9]                                           Review 5
; SYSSCHEMA=SYSDB                                    Review 6
SCHEMALIST="'DSN8910'"                               Review 7
; TABLETYPE="'TABLE','ALIAS'"                     Review 8
; CURRENTSQLID=PTG                                  Review 9

```

Note: Lines beginning with a semi-colon are comments.

Modify the items marked Review as follows:

Review 1

Specify your default Subsystem ID.

Review 2

Specify connection to only a single data base.

Review 3

Specify the subsystem to which this plan applies. There may be multiple Subsystem Sections.

Review 4

Specify the name of the plan created for this subsystem.

Review 5

Specify the data source to which the parameters that follow apply. There may be zero or more Data Source Sections.

Review 6

Used to override the default of SYSIBM for system tables.

Review 7

Used to limit the schemas that will be shown. Multiple schemas may be included in this list, each surrounded by single quotes and separated by one comma.

Review 8

Used to limit the types of object that will be shown. Any of " 'TABLE', 'ALIAS', 'VIEW', 'SYSTEM TABLE', 'SYNONYM' " may be included.

Review 9

Used to override use of the connecting user id for unqualified table names.

Optional Uncommitted Read Clause

Dynamic SQL is used to retrieve DB2 data. The SQL SELECT is internally generated and, depending on the options specified in the OPEN command, may include the FOR FETCH ONLY, ORDER BY, WHERE, and/or WITH UR clauses.

The WITH UR (Uncommitted Read) clause sets UR isolation, which can significantly improve the processing speed of DB2 data access. This setting results in reads of uncommitted data, such as data that is part of a current, partially processed transaction. If the data you are accessing is not being updated simultaneously with your Arbutus zSeries Server access, then there are no issues, even if other data in the tables is being updated.

On the other hand, if you are reading a currently active table and need to reference up to the second, but only fully completed transactions, you can disable the use of this clause. To do this, add NOUR after your DB2 plan name in the ASIINI Data Set. Separate your plan name from NOUR by one comma.

DB2 Table List

The list of tables from which the end user can select is created based on column values in the SYSTABAUTH table of GRANTEE equal to PUBLIC, PUBLIC*, the requesting User ID or the plan name. If this default is not appropriate for your site, there are a number of ways to change the selection criteria for the table list.

Displaying All Tables

To display all tables, add the keyword ALL following the DB2 plan name in the ASIINI Data Set. ALL must be separated from your plan name by one comma. If both NOUR and ALL are included, they can be entered in any order. If ALL is specified, the other methods for creating the table list described below are ignored.

Displaying Tables Based on ASIINI Criteria

The ASIINI Data Set includes an optional parameter that can be used to limit the DB2 tables that are displayed when selecting DB2 data to define by specifying DB2Tab=, followed by a fully qualified data set name, or a DDNAME. For example, the file of DB2 tables parameter in the ASIINI Data Set may look like either:

```
DB2TAB= 'SYS1.DB2TAB.LIST'
```

or

```
DB2TAB=DD:DB2TAB
```

An asterisk may be included in the fully qualified data set name. When the Arbutus zSeries Server is initialized, the asterisk is replaced with the User ID of the current user.

This data set should be created as physical sequential variable record length of maximum 255 bytes, and have the following field layout:

Start Position	Field Length	Description	Comments
1	4	DB2 subsystem name	Left justify, upper-case, pad with blanks on the right
5	1	Type of Item	T (for Table), V (for View), A (for Alias), G (for Group), C (for Custom SQL string), and must be upper-case
6	8	Owner name	When type of item is T, V, A or G: Left justify, pad

Start Position	Field Length	Description	Comments
			with blanks on the right. This value corresponds to the SYSTABAUTH column GRANTEE. When type of item is C: An SQL WHERE clause, enclosed in parenthesis, including any SYSTABAUTH columns.
14	18	Name of Item	When type of item is T, V or A: Table, view, or alias name, left justify, pad with blanks on the right

The following table illustrates examples of specifying DB2 table selection criteria using the various field layouts (described above) that are supported:

Type	Table Selection Syntax
Group	DSN1GDB2PM
Group	DSN1GDSNCLIMS
Custom	DSN1C(TCREATOR = 'Q ' AND GRANTEE = 'PUBLIC* ')
Table	DSN1TDSN8610 EMP
View	DSN1VDSN8610 PROJVIEW

When types T, V and A are included in these data sets, only the tables specified are displayed for user selection. This is useful for creating a very limited table list. Any number of type G records can be included, and all tables from all specified groups will be displayed. Only one type C record can be entered in this file, and the selection criteria can not exceed one line. A mix of selection methods (Table List (T,V,A), Groups, or custom SQL string) can be entered in these files, but only the first type encountered will be used. For example, if the first record in the file is a type G, then only Group records will be used to create the table list.

DB2 Security

Using the default DB2 security exit, all users connecting to the Arbutus Server will be assigned the DB2 authorization ID associated with the job submitter or started task. This means that the submitter or the started task owner must have sufficient authority to access the DB2 tables and views of interest to the end user. A simple method for insuring appropriate security for each end user is to run the user connections as spawned jobs. With this method, each connection is submitted by the connecting end user, and so the DB2 authorization ID is set equal to the end user's User ID.

While the DB2 security exit supplied with DB2 can be customized to enable user-level security, or the started task table can be customized to specify an owner other than the default started task owner, each site can present unique requirements and instructions for doing these things which are outside the scope of this document. Please contact Arbutus Technical Support for more information if your current DB2 security configuration may present problems for the end user.

The IMS Interface

When IMS data is processed by Arbutus zSeries Server, the data is retrieved from the hierarchical database and presented to the end-user as a flat record.

Several steps are required to allow access to IMS data:

1. Generate a Program Specification Block (PSB) for use by the program ASIMAIN, or select an existing PSB that identifies the IMS database you're interested in reading. If you're using an existing PSB, check the parameters carefully to ensure the PSB requirements described below are met.
2. Update the Arbutus zSeries Server ASIINI Data Set to include the DBD source PDS name, and a name to be used for the Arbutus zSeries Server segment table. The segment table will be created the first time the IMS interface is used.
3. Edit the IMS Session JCL IMSSESS, located in the SYSFILES PDS, and make the changes required for IMS as described below.

The Arbutus zSeries Server Program Specification Block (PSB)

The Arbutus zSeries Server requires an IMS PSB to describe the logical structure of the IMS data to be read. You may use an existing PSB, or have one generated specifically for use by the Arbutus zSeries Server.

The Program Control Block (PCB) statements for the databases to be accessed by the Arbutus zSeries Server must specify:

1. Language is ASSEMBLER (LANG=ASSEM)
2. DL/I database PCB (TYPE=DB)
3. Processing option GOP (PROCOPT=GOP)

This may include processing option T, as PROCOPT=GOTP

Option G	GET function
Option O	READ ONLY, do not en-queue to check availability
Option T	Causes DL/I to automatically retry if attempting to read data that is currently being updated by another program. If the retry fails, a GG

	status code is returned to the Arbutus zSeries Server, and the read process is terminated
Option P	Allow Path calls. Required because IMS processing includes path calls

If the PSB is being generated specifically for use by Arbutus zSeries Server, make sure that the "PSBNAME=" parameter is changed to specify a new PSB name.

Some Analyzer and ConnectPlus processes read more than one file at a time; specifically RELATIONS, JOIN and MERGE. If these will be used to read more than one IMS segment path at one time (relating data in one hierarchical path under a root key to a different path under the same key, for example) then the PSB must include a Program Control Block (PCB) for each of the different paths that may be accessed. The IMS internal control block requirements increase with each PCB defined, so only define those that are needed.

Note: When using Analyzer to relate, merge or join IMS segment paths from the same data base, if a PCB isn't available for each segment path then the message "Insufficient PCB's for multiple path processing." is displayed and the command in progress is cancelled.

In the following example, two PCBs have been created for the data base DI21PART, to allow processing of two IMS segment paths at one time:

```

PARTPSB1 PCB TYPE=DB, DBDNAME=DI21PART, PROCOPT=GOTP, KEYLEN=43, POS=M
SENSEG NAME=PARTROOT
SENSEG NAME=STANINFO, PARENT=PARTROOT
SENSEG NAME=STOKSTAT, PARENT=PARTROOT
SENSEG NAME=CYCCOUNT, PARENT=STOKSTAT
SENSEG NAME=BACKORDR, PARENT=STOKSTAT
PARTPSB2 PCB TYPE=DB, DBDNAME=DI21PART, PROCOPT=GOTP, KEYLEN=43, POS=M
SENSEG NAME=PARTROOT
SENSEG NAME=STANINFO, PARENT=PARTROOT
SENSEG NAME=STOKSTAT, PARENT=PARTROOT
SENSEG NAME=CYCCOUNT, PARENT=STOKSTAT
SENSEG NAME=BACKORDR, PARENT=STOKSTAT
PSBGEN LANG=ASSEM, PSBNAME=MULTIPLE
END

```

If multiple-file processes are run and there is no PCB available to manage processing, the message "Insufficient PCB's for multiple path processing." will be displayed and the process will be cancelled.

The POS=MULTIPLE (POS=M in the example above) parameter must be included on these PCBs for maintaining position on multiple hierarchic paths.

Updating the Arbutus zSeries Server ASIINI Data Set

The IMS interface is enabled by modifying values in the ASIINI Data Set:

```
'prefix1.ASI.SYSFILES(ASIINI)'
```

Note: The IMS Authorization Code provided in the Licensing e-mail accompanying this guide must be entered in the ASIINI Data Set. Any modifications to this value will disable the IMS interface. For more information see "IMS Authorization" on page 25.

Modify the IMS Segment Tab parameter SEGTAB= in the Arbutus zSeries Server ASIINI Data Set to specify a name to be used when creating and accessing the IMS segment table. This file contains details of the segment path built from the DBD source file. It is created the first time the IMS interface is run and read each time IMS access is requested. Once created, the segment table should be retained as a permanent data set. Depending on the standards at your site, this may require setting the expiry date.

Although we do not recommend running the IMS interface via sub-tasks (only a single connection is supported), if you do so you must provide the IMS PSB name to be used. This is specified using the IMSPSB parameter in the ASIINI Data Set. This name must be entered in upper case.

IMS Messages

Messages may be generated by the IMS control program prior to start up of the Arbutus zSeries Server. These messages are outside the control of the server, and therefore do not follow the standard Arbutus zSeries Server message convention.

In general, errors originating from the IMS control program take the form "USER ABEND nnn DEC OCCURRED PROCESSING COMMAND 'CALL'", where nnn is a three-digit system abend code.

Error Number	Description
206	The IMS DD statement in the Arbutus zSeries Server initiation CLIST is missing or invalid
662	The DLI parameter on the DFSRRC00 (DLI,ASIMAIN,PSBNAME) is incorrect
929	The PSBNAME parameter on the DFSRRC00 (DLI,ASIMAIN,PSBNAME) statement in the ASIIMS initiation CLIST does not specify a valid PSB name. The named PSB must reside in the library specified as the first entry of the IMS DD statement

Note: All other errors from the IMS control program should be reported to Arbutus Technical Support.

The Adabas Interface

The Adabas interface is enabled by modifying values in the ASIINI data set and by modifying your session JCL.

Updating the Arbutus zSeries Server ASIINI Data Set

The Adabas interface is enabled by including values in the ASIINI Data Set. All of the Adabas entries described below, if included, must be included in the [Arbutus Server Info] section of the ASIINI Data Set.

You must enter the database number (where *nn* is your database number) as:

```
ADADB=nn
```

The Arbutus zSeries Server dynamically loads the ADAUSER module when the user requests an Adabas file. Therefore, ADAUSER and its associated modules must reside in a library that the Arbutus zSeries Server can access.

If the Adabas interface module is not called ADAUSER at your site, you must include an ADAPATH entry in the ASIINI file:

```
ADAPATH=ADAINTE
```

Analyzer's Data Definition Wizard can be used to create a table layout for your Adabas file from your existing Adabas FDT's (file definition tables). To list them in the Wizard, you must edit your session JCL (see optional session JCL shown below) and include an ADAF entry in the ASIINI file:

```
ADAF=14 PAYROLL, 15 CUSTOMER, 16 INVENTORY
```

Updating the Arbutus zSeries Server Session JCL

Optional Session JCL

If your Adabas system does not run as the default SVC, you must add the following to your session JCL (where *nnn* is the SVC number that identifies your Adabas database):

```
//DDCARD DD *
ADARUN MODE=MULTI.SVC=nnn
/*
```

If you want to create table layouts in the Data Definition Wizard from existing Adabas FDT's then the FDT library must be allocated in your session JCL:

```
//ADAFDT DD DSN=HLQ.PR.DAFDT,DISP=SHR
```

System Sort

The SyncSort or DFSORT utilities can be used in place of the Arbutus Sort routine to sort many standard field types. For more information, see "System Sort Program" on page 27.

Optional sort parameters can also be added and modified as described below.

System Sort DDNames

You can supply alternate DDNAMEs for System Sort.

The Arbutus zSeries Server uses the SyncSort DDNAMEs by default. These names can be left as they are if you are running SyncSort with the standard DDNAMEs, or they can be changed to meet the requirements of your site.

If any of the DDNAMEs are omitted, placeholders of asterisks [*] will be required before any of the names that follow. Where a name is omitted, the default will be used.

The DDNAMEs must be coded in the following order, in upper case, separated by a single comma:

```
SORTDD=$ORTPARM, SORTIN, SORTOUT, SYSIN, SYSOUT
```

The Arbutus zSeries Server assumes a default parameter file DDNAME of \$ORTPARM, which is the SyncSort parameter file DDNAME. If you are running DFSORT, you must change this to DFSPARM or the sort will fail. If you are running CA-Sort, you must change this to SORTCNTL.

All of these utilities allocate the DDNAME SYSIN (or equivalent) to NULLFILE, to ensure that all sort parameters and control cards are picked up from \$ORTPARM, DFSPARM, or SORTCNTL, respectively.

Sort Parameters

You can override SORT parameters, such as MSG, using this parameter in the ASIINI Data Set. Please note that no syntax checking is done, so the parameters must be correct for the specific sort being used.

For DFSORT, you must include the word OPTION before any parameters.

The Arbutus zSeries Server will use the System Sort to sort the following key data types:

- Character
- Packed Decimal
- Binary
- Zoned
- Dates (stored as YYMMDD, YYYYMMDD, YYDDD or YYYYDDD)
- IBM Float

The System Sort is also used to sort variable-length record and VSAM data files. When the input file is VSAM, the output file is created as a variable-length record blocked data set, with the record length equal to the maximum record length of the input file. The Arbutus Sort routine cannot handle variable-length

or VSAM files. Data sets with undefined record lengths cannot be sorted by either System Sort or the Arbutus Sort.

The System Sort may allocate temporary DASD or virtual storage depending on the sort requirements. Sort work files may be allocated in the Arbutus zSeries Server initiation CLIST, DDNAME SORTWORKnn, where nn is a number from 01 to 32. A minimum of six sort work files is recommended. If sort work files are not allocated in the initiation CLIST, they will be dynamically acquired, if needed, when the SORT command initiates system sort.

If the Arbutus Sort routine is invoked, the Arbutus zSeries Server displays the message "ASI SORT applied due to type or complexity of the sort key."

All sort routines are performed as stable sorts. This means that records with the same key value will retain the same relative position in the output file.

CHAPTER 4 ZSERIES SERVER SYSTEM MESSAGES

"S0010 Unable to open the INI file."

The server initialization routine has failed while attempting to open the file associated with DDNAME ASIINI in the server initiation CLIST. This error occurs if the INI Data Set has not been allocated to the current TSO session associated with DDNAME ASIINI.

Verify that the data set name specified in the server initiation CLIST for DDNAME ASIINI has been entered correctly. This file is the data set 'prefix.asi.asiini' copied during the installation procedure. If the prefix selected for this data set during installation is different than the prefix associated with the current TSO session, the name must be fully qualified.

"S0011 Unable to open ASISYS01"

Offline execution JCL has failed. Arbutus zSeries Server was unable to open the data set specified for step ASISYS01.

"S0012 Must specify default prefix/userid"

Offline execution JCL has failed. The prefix or User ID has not been specified in ASISYS01.

"S0013 Prefix in ASISYS01 is too long; maximum 31"

Offline execution JCL has failed. The prefix specified in ASISYS01 exceeds 31 characters.

"S0014 TSO userid is too long; maximum 8"

Offline execution JCL has failed. The User ID specified in ASISYS01 exceeds eight characters.

"S0020 SYSTEM FILE PDS must be specified."

The system file PDS parameter specified in the ASIINI Data Set is blank. Edit the 'prefix1.ASI.SYSFILES', member ASIINI copied during installation and enter the data set name of the System PDS ('prefix.asi.sysfiles') copied during installation.

"S0040 Unable to open command file command.file.name."

The initialization routine has failed while attempting to open member @ASICMD in the System file PDS. This error occurs if member @ASICMD has been deleted.

Restore member @ASICMD from a current backup of the Arbutus system or copy it from the installation tape.

"S0045 Unable to read command file."

The server initialization routine has failed while attempting to read member @ASICMD from the System file PDS. This can only occur if the contents of member @ASICMD have been deleted. Restore @ASICMD from a current backup of the Arbutus system or copy this member from the installation tape.

"S0050 Default DBD Source PDS must be specified"

The ASIINI Data Set contains a valid IMS enable code, but IMS DBD PDS parameter is blank. Enter the name of the DBD Source code PDS in the ASIINI Data Set. If the IMS interface will not be used, any text string can be entered.

"S0060 IMS Segment Table must be specified"

The ASIINI Data Set contains a valid IMS enable code, but the IMS Segment Tab parameter is blank. Enter any name for the Segment Table to be created in the ASIINI Data Set. If the IMS interface will not be used, any text string may be entered.

"S0070 Unable to allocate default storage, allocating x bytes."

This is an information only message. The server initialization routine has failed while attempting to allocate the default work space for the application. The number of bytes to be allocated is specified in the ASIINI Data Set.

The message displays the amount of space that was actually acquired. If the default amount of work space is routinely unavailable, the ASIINI Data Set default should be modified to a more appropriate value, or the region size allocated for the TSO session should be increased.

"S0075 Failed allocating x bytes of virtual storage."

This message is always preceded by message S0070. The server initialization routine has failed while attempting to allocate half of the original work space requested. The region size allocated for the current TSO session should be increased, or the default work space value specified in the ASIINI Data Set should be decreased.

"S0080 ASI LICENSE has expired."

The Arbutus zSeries Server software expiration date has been reached. Contact Arbutus Technical Support for assistance.

"S0085 Serial Number is invalid."

The serial number specified in the ASIINI Data Set is either blank, or longer than four characters. Contact Arbutus Technical Support for assistance.

"S0088 CPU ID is invalid."

The CPU ID specified in the ASIINI Data Set is not valid. Contact Arbutus Technical Support for assistance.

"S0090 Expiry Date is invalid"

The expiry date specified in the ASIINI Data Set is not valid. Contact Arbutus Technical Support for assistance.

"S0095 CPU nnnnnn-xxxx does not match INI file entry."

The machine is not licensed to run Arbutus zSeries Server. Contact Arbutus Technical Support for assistance.

"S0105 - Failed checking concurrent user code"

An error has occurred while checking the concurrent user code specified in the ASIINI Data Set. If the concurrent user number, prefix, or code has been changed, restore these as they appeared in the ASIINI Data Set when it was originally installed. If these codes have not been changed call Arbutus Technical Support.

"S0115 - Maximum ASI Users logged on"

This installation of Arbutus zSeries Server has been licensed for a maximum number of users, and that maximum has been reached. Try logging on again later.

"S0125 - Concurrent user information missing"

An error has occurred while checking the concurrent user codes specified in the ASIINI Data Set. Restore the codes as they appeared in the ASIINI Data Set when originally installed.

"S0136 - Name with substitution is too long (file-name)"

Using asterisk [*] to substitute userid in file-name, and resultant file name is too long. Edit ASIINI Data Set and correct name.

"S0140 - 'name' is invalid; level or member may not begin numeric"

Cannot start a level name or member name with a number.

"S0145 - 'name' is invalid; level or member is longer than 8 chars"

Level name or member name must be eight characters or less.

CHAPTER 5 OFFLINE INTERFACE

Overview

The Arbutus zSeries Server can be run offline using JCL provided with this installation package. This chapter describes the steps required to install this interface.

System Requirements

This section describes additional ASIINI Data Set and Offline Execution JCL modifications that may be required when running the offline interface.

The ASIINI Data Set

The Configuration file and the Format PDS (for table layouts) can be modified by the user. It may be appropriate to provide each user, or group of users, with individual copies of these files.

Default Table Layout PDS

This PDS can be used to store table layouts that describe the data files accessed by the user when running with the offline interface.

Use the ASIINI parameter `FORMAT=` to change the default format location from `DDNAME FORMAT`.

To specify an alternate partitioned data set, enter the fully qualified name all in uppercase, as:

```
FORMAT= ' ANALYZ . FMT . PDS '
```

To specify that table layouts should be stored as sequential files, enter the keyword `NOPDS` all in uppercase, as:

```
FORMAT=NOPDS
```

The `SET FORMAT NOPDS` command can also be used to achieve this.

If you do not override the default of `DDNAME FORMAT`, this DD name and the associated PDS must be added to your execution JCL and CLIST.

Configuration File

Contains commands that are executed each time that Arbutus Analyzer is run using the offline interface.

Use the ASIINI parameter `CONFIG=` to change the default configuration location from `DDNAME CONFIG`.

If you do not override the default of `DDNAME CONFIG`, this DD name and the associated data set must be added to your execution JCL and CLIST.

Edit the Offline Execution JCL

Edit member ASIBATCH (or IMSBATCH if accessing IMS) in the SYSFILES PDS '*prefix1.ASI.SYSFILES*' to reflect the data set names installed at your site:

```
//ASIBATCH JOB (), 'ASI BATCH',MSGCLASS=D,TIME=NOLIMIT
//STEP01 EXEC PGM=ASIMAIN,DYNAMNBR=35,REGION=8192K
//STEPLIB DD DSN=prefix1.ASI.LOADLIB,DISP=SHR
//ASIINI DD DSN=prefix1.ASI.SYSFILES (ASIINI),DISP=SHR
//MVSIN DD DSN=prefix1.ASI.SYSFILES (ASITEST),DISP=SHR
//ASISYS01 DD *
prefix TSOID
/*
//SYSTSPRT DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//
```

The DD MVSIN names a data set that you create, containing the Analyzer commands to be run offline.

The DD ASISYS01 specifies the prefix and User ID under which this offline session should be run.

Change the prefix to the user's high-level TSO prefix and the TSOID to the user's User ID.

Note: Both the prefix and User ID must be specified, even if they are identical.