



Informatica® SSA-NAME3
10.5

SSA-NAME3 Installation Guide

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2023-09-28

Table of Contents

Preface	5
Learning About Informatica SSA-NAME3.	5
What Do I Read If.	7
Informatica Resources.	8
Informatica Network.	8
Informatica Knowledge Base.	8
Informatica Documentation.	8
Informatica Product Availability Matrices.	8
Informatica Velocity.	9
Informatica Marketplace.	9
Informatica Global Customer Support.	9
 Chapter 1: SSA-NAME3 Installation.....	10
Installation Overview.	10
Installation Sequence	10
 Chapter 2: Post Installation Steps (Windows).....	11
Overview.	11
Getting Started.	11
Connection from Windows to Server Platform(s).	11
Developer's Workbench.	12
The Developer's Workbench.	12
Accessing the Online Documentation.	12
 Chapter 3: Post-Installation Steps (UNIX).....	14
Overview.	14
Getting Started.	14
Configure a Web-Browser.	14
Standard Populations.	14
Developer's Workbench.	15
Development and Testing.	15
Production Use.	15
 Chapter 4: z/OS Installation.....	16
Installation Overview.	16
Minimum Requirements.	16
Population Override Manager and Edit Rule Wizard Limitations.	17
Testing Overview.	17
SSA-NAME3 Release Structure.	18
Installing SSA-NAME3.	18

Step 1 - Copy the Files from the CD to a local HDD.	18
Step 2 - Transfer the Datasets.	18
Step 3 - Copy the Required CICS Modules.	21
Step 4 - Customize the SSA-NAME3 Server JCL.	22
Transferring Custom Population Files.	23
Testing SSA-NAME3.	24
Testing the Batch Cobol Sample.	25
Testing the Batch PL/I Sample.	25
Testing the CICS Cobol Sample.	26
Appendix.	28
Copybooks.	28
Possible Problems.	29
Program Definitions.	29
Additional entries for Natural.	30
Transaction Definition.	33
Other Cobol Notes.	33
Other PL/I Notes.	33
Natural.	34
Contents of Release Datasets.	34
Populations available.	36
Populations requiring CJK license.	37
Index.	38

Preface

Welcome to the Informatica SSA-NAME3 Installation Guide. This guide is intended to be the first technical material a new user reads before installing or using the SSA-NAME3 software.

This guide describes the post-installation steps for Windows/Unix systems.

A user would install on a Windows either because the search application which will use SSA-NAME3 runs in that environment, or because they wish to use one of the SSA-NAME3 clients on the Windows, even though the search application itself may run on a different platform.

For UNIX users, this guide shows how to install the software and how to either connect to it from a Workbench running on a Windows computer, or how to install the Workbench natively on the Unix system.

Learning About Informatica SSA-NAME3

This section provides details of documentation available with the SSA-NAME3 product.

Introduction to SSA-NAME3

Provides an overview of SSA-NAME3. It is written in a way that can be read by someone who has no prior experience of the product and wants a general overview of SSA-NAME3. It explains the problems SSA-NAME3 overcomes and provides an overview of how this is done. One chapter is dedicated to providing an overview for Application Programmers.

Getting Started

This manual is intended to be the first technical material a new developer or designer reads before installing or using the SSA-NAME3 software, regardless of the platform or environment. Its goal is to help a new user get the software installed and produce a working prototype application that calls SSA-NAME3 and executes searches against their own data.

To achieve this it provides a "script" to follow which includes pointers to pertinent sections of the other manuals.

Application & Database Design

This manual contains tips and techniques useful for setting up and optimizing a name search and matching application, including database issues, and illustrates best-practice techniques, common pitfalls, and strategies regarding the subject of name and address matching.

Installation Guide

This manual provides information on how to install the SSA-NAME3 product.

SSA-NAME3 Workbench User Guide

This is a guide to using the SSA-NAME3 Workbench - a Java GUI tool that helps a programmer understand and prototype SSA-NAME3 calls. The Workbench is also used for:

- Generating Sample Program Code;
- Executing SSA-NAME3 Calls;
- Testing different SSA-NAME3 run-time options;
- Producing debugging and support information for Informatica Corporation

Note: The Workbench in itself is not a search and match application. It assists the developer build a search and match application.

API Reference

The ultimate goal of an SSA-NAME3 implementation is for application programs to be able to call SSA-NAME3's API Functions to build keys and search strategies and to compute match scores and decisions.

This manual describes a typical program process flow for building an identity search application, and also lists in detail each of the API Functions. It describes the parameters required by these functions and the information returned.

Population Override Manager User's Guide

This is a guide to using the SSA-NAME3 Population Override Manager - a Java GUI tool that allows a trained data analyst to override some of the Standard Population rules that are supplied with the product, or provided in the form of a Custom Population. The types of rules that can be overridden using this tool are:

- Edit-list rules
- Frequency tables
- Scalar Frequency Tables
- Matching Purposes

Note: Use of this tool without proper training from Informatica should not be attempted, as improper use can adversely affect the reliability and performance of the search application(s).

Edit Rule Wizard User's Guide

This is a guide to using the SSA-NAME3 Edit Rule Wizard - a Java GUI tool that helps a business user safely add certain types of Edit Rules to the Standard or Custom Population without requiring specific knowledge of SSA-NAME3 or support from a programmer or data analyst. The types of rules that can be added using this tool are:

- Discard a word or phrase when searching and matching (e.g. a new "noise" word)
- Add a new replacement word or phrase when searching and matching (e.g. a new "abbreviation", "nickname" or "acronym")
- Add a new compound name marker word

Release Notes

The Release Notes contain information about what's new in this version of SSA-NAME3. It is also used to summarize any documentation updates as they are published.

What Do I Read If. . .

I am. . .

. . . a business manager

The INTRODUCTION TO SSA-NAME3 will address questions such as "Why have we got SSA-NAME3?", "What does SSA-NAME3 do"?

I am. . .

. . . a system designer or DBA

The INTRODUCTION TO SSA-NAME3 will address questions such as "What resources are needed to implement SSA-NAME3?". The APPLICATION & DATABASE DESIGN manual will lead you through many of the design considerations of name search and matching applications.

I am. . .

. . . installing SSA-NAME3

Before attempting to install SSA-NAME3 you should read the Getting Started document. This will describe the pre-requisites and help you plan the installation and implementation of SSA-NAME3. The actual installation steps for your platform are documented in the Installation Guide.

I am. . .

. . . an Analyst or Application Programmer

A high-level overview is provided specifically for Application Programmers in the INTRODUCTION TO SSA-NAME3 manual. Before attempting to develop programs that interface with SSA-NAME3, you should also read the GETTING STARTED and APPLICATION & DATABASE DESIGN manuals, as well as experimenting with calls in the WORKBENCH USER GUIDE.

When developing the application program(s), use the API REFERENCE manual which describes a typical application and the Function parameters.

Working example programs that illustrate the calls to SSA-NAME3 in various languages are available by using the Sample Program button on the Workbench.

I want to know. . .

. . . what SSA-NAME3 does

The INTRODUCTION TO SSA-NAME3 manual gives an overview of what SSA-NAME3 does and how it does it.

I want to know. . .

. . . how to setup the database

Refer to the APPLICATION & DATABASE DESIGN manual for tips and techniques on configuring the database to store SSA-NAME3 Keys and optimizing it for searching and matching.

I want to know. . .

. . . how to code a search application

The INTRODUCTION TO SSA-NAME3 manual contains a specific section designed to get application programmers familiar with the concepts of developing an SSA-NAME3 search and match application.

The API REFERENCE GUIDE details the Function calls required and their parameters. The SSA-NAME3 WORKBENCH USER GUIDE shows how to generate a sample program in a variety of programming languages.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

CHAPTER 1

SSA-NAME3 Installation

This chapter includes the following topics:

- [Installation Overview, 10](#)
- [Installation Sequence, 10](#)

Installation Overview

This topic provides information about SSA-NAME3 installation.

How SSA-NAME3 is Delivered

SSANAME3 consists of the SSA-NAME3 Callable Routine, standard populations, SSA-NAME3 Workbench, Population Override Manager, and Edit Rule Wizard.

Customers who evaluate and license SSA-NAME3 receive a download link for the installation media from Informatica Shipping. Customers who evaluate the product can also receive a 30-day temporary license key, which allows for a full installation of SSA-NAME3 with no restrictions.

Use the Informatica installer to install on Windows or UNIX platforms. You can run the installer in the graphical mode or in the console mode. To install SSA-NAME3 on z/OS, use the `.zip` file that contains a number of load libraries.

Choices to Be Made Before Installation

If you upgrade SSA-NAME3, the installer overwrites the standard populations, so back up your active populations before you upgrade.

Installation Sequence

The 'target' computer is the computer where the SSA-NAME3 Callable Routine and Standard Populations are installed. Client components, consisting of the Developer's Workbench, Population Override Manager and Edit Rule Wizard are also installed on the target computer, but may be installed on other client computers as well.

The SSA-NAME3 product is installed using the Informatica Product Installer provided with the installation media.

If you plan to install client components on other computers, you must first complete installation on the target computer and then install on the client computers providing the hostname for the target computer. Before the clients can be used, the SSA-NAME3 server must be started on the target computer.

CHAPTER 2

Post Installation Steps (Windows)

This chapter includes the following topics:

- [Overview, 11](#)
- [Connection from Windows to Server Platform\(s\), 11](#)
- [Developer's Workbench, 12](#)
- [The Developer's Workbench, 12](#)

Overview

This chapter describes the post-installation procedure for SSA-NAME3 on the MS Windows platforms.

Getting Started

You need to read the Getting Started guide before using the product. In particular, the following section contain useful information:

- An Overview of the SSA-NAME3 Components
- Implementation Architecture
- Resources Required to Implement SSA-NAME3

Connection from Windows to Server Platform(s)

A TCP/IP connection is required if the Population Override Manager or Edit Rule Wizard are to be used (because these clients communicate only with the SSA-NAME3 Server), or if the Developer's Workbench is to run on a client and connect to the SSA-NAME3 Server on a remote computer.

An application can also access the SSA-NAME3 Server using this network connection. However it is not recommended that remote calls are used in a production system due to performance degradation.

Developer's Workbench

The SSA-NAME3 Developer's Workbench can be used to experiment with the API Functions by simulating an application workflow and parameters. The Workbench also provides access to the full documentation set.

The Developer's Workbench

The Developer's Workbench can be started from the Start Menu. Use the icon for the DLL version if the Workbench and Callable Routine are on the same computer. If the Callable Routine is installed on a remote computer, then it should be started as the SSA-NAME3 Server on that computer before continuing. Once the SSA-NAME3 Server is running, it can be connected to using the socket version icon.

The purpose of the Developer's Workbench is to provide a simulation environment to perform function calls in exactly the same way your application program does and also to provide access to the SSA-NAME3 Documentation.

The initial screen is as follows:



Accessing the Online Documentation

This entry screen provides access to the online documentation. These HTML documents are best viewed with Microsoft Internet Explorer Version 5 or above, or Netscape Version 5 or above.

It is recommended that a new user read the *INTRODUCTION TO SSA-NAME3 GUIDE* before starting. This will help with an understanding of the purpose and functions of the product.

For more detail regarding the use of the Developer's Workbench, refer to the *WORKBENCH USER GUIDE*. For information regarding Application and Database design, read the *APPLICATION and DATABASE DESIGN*

GUIDE. For more detail regarding the API Functions and required parameters to use in your programs, refer to the *API REFERENCE GUIDE*.

CHAPTER 3

Post-Installation Steps (UNIX)

This chapter includes the following topics:

- [Overview, 14](#)
- [Development and Testing, 15](#)
- [Production Use, 15](#)

Overview

This guide describes the post-installation procedure for SSA-NAME3 in a Unix environment, including AIX, HP/UX, Linux, Solaris and z/OS Unix System Services.

Getting Started

You need to read the Getting Started guide before using the product. In particular, the following sections contain useful information:

- An Overview of the SSA-NAME3 Components
- Implementation Architecture
- Resources Required to Implement SSA-NAME3

Configure a Web-Browser

On UNIX platforms, edit `$$SABIN/ssabrz` and uncomment one of the statements that sets the environment variable `SSABRZ` to an appropriate Web browser.

Standard Populations

The Standard Populations are installed by the Informatica IR Product Installer. They reside in the `$$SSATOP/pr/default` directory, where `$$SSATOP` is the root of the installation.

Moving populations between Unix and Windows Installations

The same directory structure is used on the Windows release. If populations are to be transferred between Unix and Windows, ftp the required populations in ascii mode to the corresponding directory on the other machine.

Developer's Workbench

The SSA-NAME3 Developer's Workbench can be used to experiment with the API Functions by simulating an application workflow and parameters. The Workbench also provides access to the full documentation set.

Development and Testing

To start the SSA-NAME3 Server for access by remote development clients and testing, use the command:

```
$$SSABIN/n3up
```

The Developer's Workbench, Population Override Manager and Edit Rule Wizard, and optionally an application program while in development and testing, can now connect to the server across the network or locally.

Note: The Population Override Manager and Edit Rule Wizard require the server, even if running locally. The Workbench and an application program can use a DLL/Shared Library if running locally. Refer to the client user guides and the *Application and Database Design* guide for more information.

To bring the server down, use the command:

```
$$SSABIN/n3down
```

An application on the same UNIX computer may also call the SSA-NAME3 Callable Routine through the server, or call it locally through the shared library.

To be able to call the SSA-NAME3 shared library from your application, the following environment variable must be set up:

```
SSAPR=$SSATOP/pr; export SSAPR
```

This points to the directory where the Population Rule sets (the .ysp files) are to be found. \$SSATOP/pr is the default location. If this is changed, change the SSAPR variable appropriately.

Also, ensure that the \$\$SSABIN directory is included in the LD_LIBRARY_PATH (or equivalent). For example,

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$$SSABIN;  
export LD_LIBRARY_PATH
```

The \$\$SSABIN path must also be added to the PATH variable: PATH=\$PATH:\$\$SSABIN; export PATH

Licensing Note

The use of the SSA-NAME3 CJK-SUPPORT (Chinese, Japanese and Korean double-byte) and the Arabic Mixed populations requires a separate license. Contact your local Informatica Corporation support office for details.

Production Use

A production application may also call the SSA-NAME3 Callable Routine through the SSA-NAME3 Server, or through the shared library.

Applications that use the shared library may be more efficient; however, in some environments it may be easier to develop using the server.

To use the SSA-NAME3 Server requires that it be started and running in the production environment. Therefore, the shared library may be operationally easier to maintain.

You need to call your Informatica Corporation technical consultant to discuss the options.

CHAPTER 4

z/OS Installation

This chapter includes the following topics:

- [Installation Overview, 16](#)
- [SSA-NAME3 Release Structure, 18](#)
- [Installing SSA-NAME3, 18](#)
- [Transferring Custom Population Files, 23](#)
- [Testing SSA-NAME3, 24](#)
- [Appendix, 28](#)

Installation Overview

The SSA-NAME3 mainframe release is typically delivered on a CD as a zip file containing a number of library datasets. Refer to the *SSA-NAME3 Release Structure* section below for more details on what the libraries contain.

Following is an outline of the steps needed to install SSA-NAME3 on the mainframe:

- Unzip the install file from the CD to a local HDD
- Transfer the unzipped files from the local HDD to Mainframe datasets

The following is an outline of the extra install steps required to support CICS applications that will call SSA-NAME3, rather than invoke a database stored procedure that calls SSA-NAME3.

- Copy the required SSA-NAME3 modules into a CICS Load Library and define them to CICS.
- Customize the JCL used to bring the SSA-NAME3 Server up and down (N3UP, N3DOWN).

These steps are detailed in the Installing section.

Minimum Requirements

This section describes the minimum requirements to install SSA-NAME3.

All Users

Whether your SSA-NAME3 application will run in batch or under CICS, you must have at least the following:

- z/OS V1.2 or later
- Language Environment installed and enabled

CICS Applications

There are a couple of ways that CICS applications can call on SSA-NAME3 services.

If the database being used supports multi-threaded stored procedures, the applications that call SSA-NAME3 may be implemented as database stored procedures that dynamically call the SSA-NAME3 load module.

Alternatively, if not using database stored procedures, i.e. CICS applications will call SSA-NAME3 at an "application" level, then either the SSA-NAME3 Server must be used and started in a separate region or the DLL version of the population must be used. CICS applications communicate with the SSA-NAME3 Server through a SSA-NAME3 stub program that uses sockets and TCP/IP. The additional requirements for CICS applications calling SSA-NAME3 and application level are therefore:

- CICS TCP/IP Socket Interface support must be installed, configured and enabled;
- Language Environment (LE) must be enabled in CICS.

Note: Both CICS TCP/IP Socket support and the Language Environment are components that should be installed and/or verified by a systems programmer.

The USERID that Submits the SSA-NAME3 Server

Batch SSA-NAME3 applications may also use the SSA-NAME3 Server, or they can load the SSA-NAME3 Callable routine into the batch region.

If the Developer's Workbench client is to be used to access the SSA-NAME3 callable routine running under z/OS, the SSA-NAME3 Server must be started.

If the SSA-NAME3 Server is to be used, then the USERID that submits the SSA-NAME3 Server JCL must have an OMVS (USS) segment defined. This is required to add permission to access TCP/IP into RACF.

Population Override Manager and Edit Rule Wizard Limitations

The Population Override Manager and Edit Rule Wizard can not currently be used with an SSA-NAME3 Server that is running under IBM z/OS. However, they can be used with an SSA-NAME3 Server that is running under Unix System Services on the z/OS mainframe.

Testing Overview

Sample programs are provided for Cobol and PL/I.

The following is an outline of the steps to compile and test a batch program.

- Customize the JCL that compiles and runs the sample batch program
- Compile and link the sample batch program
- Run the program and check the results

The following is an outline of the steps to compile and test a CICS program that will use the SSA-NAME3 Server.

- Customize the JCL provided that compiles and runs the sample CICS program
- Compile and link the sample CICS program and copy it to a CICS Load Library
- Create a CICS transaction to run the sample program
- Run the JCL to bring the SSA-NAME3 Server up
- Run the transaction and check the results
- Run the JCL to bring the SSA-NAME3 Server down

These steps are detailed in the Testing section.

SSA-NAME3 Release Structure

The SSA-NAME3 release contains the following types of datasets.

Load datasets

The Load datasets contain:

- Load modules for the core SSA-NAME3 modules
- Load modules for the SSA-NAME3 Server application
- Special load modules for PL/I dynamic calls

Source dataset

The Source dataset contains:

- JCL to start and stop the SSA-NAME3 Server
- Sample Cobol and PL/I source code
- JCL to compile and run the sample programs

Population datasets

For each supported Population the SSA-NAME3 Standard Population Rules are supplied in 2 forms, as a .ysp file and as a load module. The datasets containing the .ysp files have the prefix `N3V100.PR`, while the datasets containing the load modules have the prefix `N3V100.DLL`. Users may use either ysp or dll files.

Note: Use of the SSA-NAME3 CJK-SUPPORT double-byte populations (Chinese, Japanese & Korean), and the Arabic Mixed population, requires a separate license. Contact your local Informatica Corporation support office for details.

Installing SSA-NAME3

The following are detailed steps for installing SSA-NAME3 on the mainframe.

Step 1 - Copy the Files from the CD to a local HDD

Copy the required datasets from the CD to your local HDD (optional).

The `N3V100.PLI.LOAD` dataset can be omitted if you will not be developing PL/I applications to call SSA-NAME3.

Step 2 - Transfer the Datasets

The SSA-NAME3 datasets are supplied in TSO TRANSMIT format. They can be transferred to the mainframe using either the `IND$FILE` transfer method, or if z/OS FTP Server is available, using FTP.

A high-level dataset name qualifier will need to be set up on the mainframe, accessible by the TSO User-id that will do the transfer. In the examples below, this is `IDS`.

The target datasets will only be temporary datasets, as they still must be converted to z/OS PDS datasets by the receive command in a TSO session. Therefore, it is a good idea to name them in a way that will not conflict with the names you will ultimately assign. In the examples below, the word `XMIT` has been added to the intermediate datasets for this purpose.

FTP Transfer

The following section provides examples of FTP to mainframe using Win32 FTP client:

Transfer the populations:

```
ftp> quote site lrecl=80 recfm=fb blksize=3120 vol=ssawrk cylinders primary=150  
secondary=10  
ftp> bin  
ftp> put n3 v100.pr.default 'nm3.xmit.n3 v100.pr.default'
```

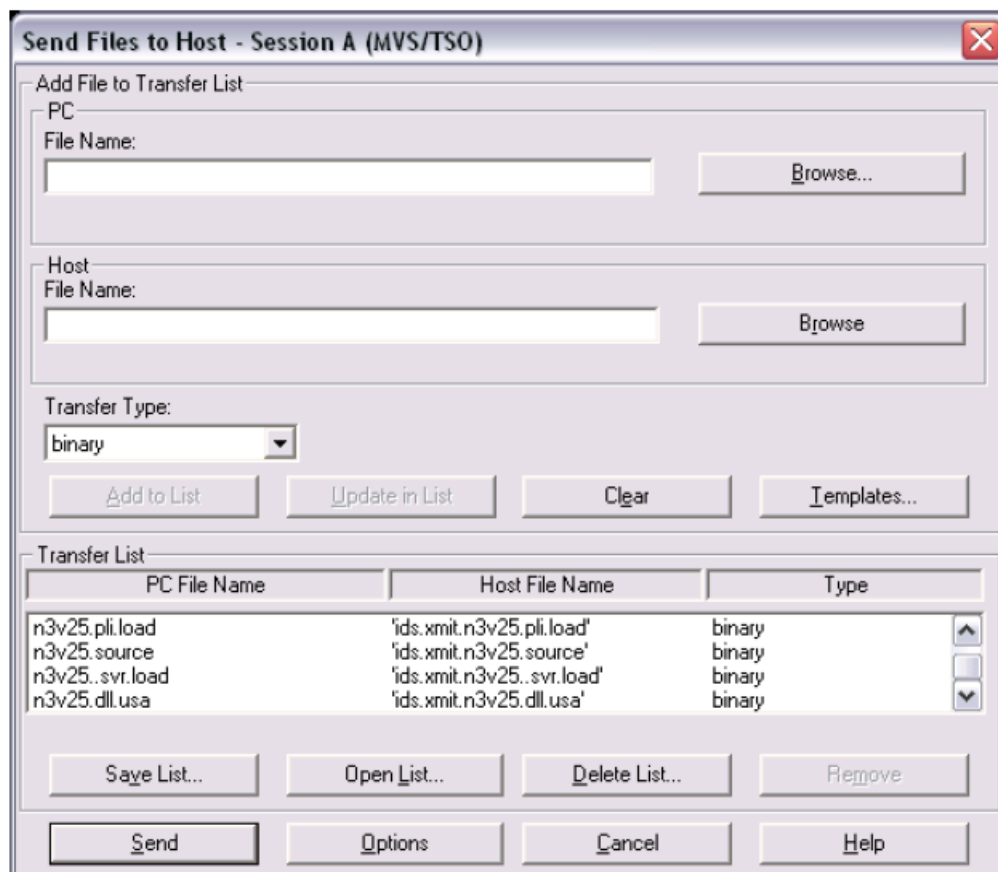
Transfer the other files:

```
ftp> quote site lrecl=80 recfm=fb blksize=3120 vol=ssawrk cylinders primary=5 secondary=1  
ftp> bin  
ftp> put n3 v100.source 'nm3.xmit.n3 v100.source'  
ftp> put n3 v100.core.load 'nm3.xmit.n3 v100.core.load'  
ftp> put n3 v100.svr.load 'nm3.xmit.n3 v100.server.load'  
ftp> put n3 v100.pli.load 'nm3.xmit.n3 v100.plidyn.load'
```

Note: The datasets must be transferred in binary mode.

IND\$FILE Transfer

To transfer using the IND\$FILE method, use an appropriate IND\$FILE transfer application, for example:



The IND\$FILE transfer screen is typically provided with your 3270 emulator.

Ensure to check the following when transferring:

- All datasets must be transferred in Binary mode.

- Typically, use single quotes around the target dataset name to avoid your User-id being appended to the beginning of the name on the mainframe.
- Before sending the files, ensure you are at the TSO command prompt on the mainframe (either native TSO, or ISPF option 6).
- For all datasets, specify at least the following parameters to the `Send File` function:
`RECFM(F)`
`LRECL(80)`

For `CORE.LOAD` and `SVR.LOAD`, also specify the `SPACE` parameter:

`SPACE(5,1) CYLINDERS`

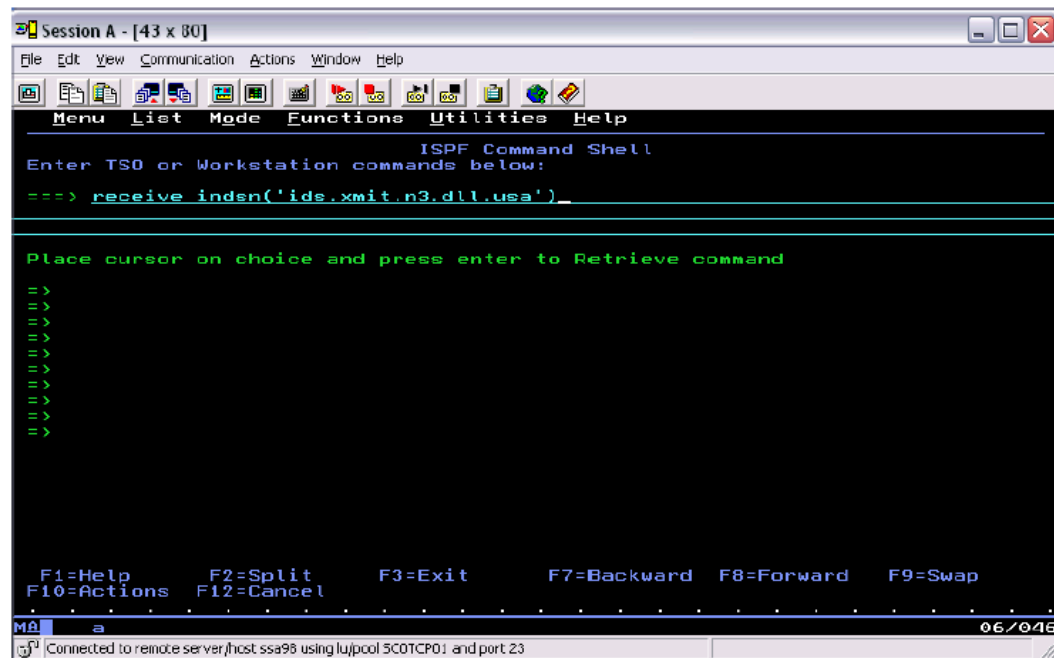
Depending on the package used, the above `IND$FILE` parameters may either be available as GUI controls on the `IND$FILE` transfer screen or they may need to be specified in a free-form additional parameters field.

Note: You need to transfer all required Standard Population datasets. In the example above, the USA populations is being transferred. Also, note that only the DLL (dataset prefix `N3V100.DLL`) or the ysp (dataset prefix `N3V100.PR`) needs to be transferred, not both.

Receiving the Files

The uploaded files must be "received" using the TSO RECEIVE command. This is regardless of whether they were sent through `IND$FILE` or FTP. This "receive" will convert the datasets from TSO/E Transmit format to PDS datasets.

The TSO RECEIVE command is executed from the TSO Command section of ISPF. This is normally option 6 on the ISPF main menu.



An example for TSO RECEIVE command:

```
receive indsn('nm3.xmit.n3 v100.core.load')
```

1. At the prompt enter: `da('nm3.n3 v100.core.load')` or `da('nm3.n3 v100.core.load') volume(xxxxxx)`
2. Press Enter to restore load library.

It is also possible to receive the datasets through JCL. For example,

```
//SSAN3INS JOB (1),IEBCOPY,NOTIFY=&SYSUID
//*****
//*
//*  INSTALL SSA-NAME3 V100 XMIT DATASETS
//*
//*****
//STEP0 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEL SSA.N3V100.SERVER.LOAD
    DEL SSA.N3V100.CORE.LOAD
    DEL SSA.N3V100.PLIDYN.LOAD
    DEL SSA.N3V100.SOURCE
    DEL SSA.N3V100.PR.DEFAULT
/*
//*
//STEP1 EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDSN('SSA.XMIT.N3V100.SVR.LOAD')
DA('SSA.N3V100.SERVER.LOAD') VOLUME(SSAWRK)
/*
//*
//STEP2 EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDSN('SSA.XMIT.N3V100.CORE.LOAD')
DA('SSA.N3V100.CORE.LOAD') VOLUME(SSAWRK)
/*
//*
//STEP3 EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDSN('SSA.XMIT.N3V100.PLI.LOAD')
DA('SSA.N3V100.PLIDYN.LOAD') VOLUME(SSAWRK)
/*
//*
//STEP4 EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDSN('SSA.XMIT.N3V100.SOURCE')
DA('SSA.N3V100.SOURCE') VOLUME(SSAWRK)
/*
//*
//STEP5 EXEC PGM=IKJEFT1A
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDSN('SSA.XMIT.N3V100.PR.DEFAULT')
DA('SSA.N3V100.PR.DEFAULT') VOLUME(SSAWRK)
/*
//
```

Note: Delete the temporary transfer datasets once you have finished the receives.

Step 3 - Copy the Required CICS Modules

This section describes about the CICS modules to be copied.

CICS Applications that will call the SSA-NAME3 Server

If you are a CICS user who will be developing applications that will call the SSA-NAME3 Server (rather than using a database stored procedure or a Population DLL), then perform the steps provided below:

1. Copy the modules *CSSA/OK*, *SSALSN*, *SSAN3CC*, *SSAOPENC*, *SSAN3FN* from *SSA.N3V100.CORE.LOAD* to a *LOAD* library referenced from the CICS *DFHRPL*.

2. Optionally, create your own dataset to copy the modules into. This dataset must then be specified in the DFHRLP DD card of the CICS start-up job. For example, create a dataset called SSA.N3V100.CICS.LOAD (using the same data set attributes as for SSA.N3V100.CORE.LOAD) and copy the above members into this dataset.
3. After the SSA-NAME3 CICS modules have been copied, rename the module SSAOPENC to SSAOPEN .

Using a group defined on your CICS system, define the programs to CICS. For example, use the CEDA commands:

```
CEDA DEF PROGRAM(SSAN3FN) GROUP(IDSGROUP)
CEDA INS PROGRAM(SSAN3FN) GROUP(IDSGROUP)
```

Repeat for the other three programs. The actual definitions is detailed in the Appendix.

CICS Applications that will use Population DLLs

If you are a CICS user who will be developing applications that will use SSA-NAME3 Population DLLs (rather than a database stored procedure or calling the SSA-NAME3 Server), then perform the steps provided below:

1. Copy the modules SSAIOK, SSALSN, SSAN3FN, SSAN3TB, SSAN3V2, SSAN3CL and SSAOPEN from SSA.N3V100.CORE.LOAD to a LOAD library referenced from the CICS DFHRLP.
2. Optionally, create your own dataset to copy the modules into. This dataset must then be specified in the DFHRLP DD card of the CICS start-up job). For example, create a dataset called SSA.N3V100.CICS.LOAD (using the same data set attributes as for SSA.N3V100.CORE.LOAD) and copy the above members into this dataset.
3. Copy the required populations from the SSA.N3V100.PR.DEFAULT.LOAD dataset to a LOAD library referenced from the CICS DFHRLP. For example, to use the Australia population copy the module AUSTRALI.

Using a group defined on your CICS system, define the programs to CICS. For example, use the CEDA commands:

```
CEDA DEF PROGRAM(SSAN3FN) GROUP(IDSGROUP)
CEDA INS PROGRAM(SSAN3FN) GROUP(IDSGROUP)
```

Repeat for the other three programs. The actual definitions is detailed in the Appendix.

Utilising the Population DLLs at run time

In order for the Populations DLLs to be used by SSA-NAME3, it is necessary to add the control LOADDLL=Y to the controls parameter passed to the open call. If this is not done, then SSA-NAME3 will attempt to load the YSP and the DLL will be ignored.

Step 4 - Customize the SSA-NAME3 Server JCL

If you plan to develop applications that call SSA-NAME3 through the SSA-NAME3 Server, you must customize the SSA-NAME3 Server JCLs.

Customize the following JCL that starts the SSA-NAME3 Server in the SSA.N3V100.SOURCE(N3UP) member:

```
//SSAN3UP JOB , 'N3UP', REGION=150M, NOTIFY=&SYSUID
//SSAN3SV EXEC PGM=SSAN3SV,
// PARM=('ENVAR("SSAPR=SSA.N3V90.PR")'),
// '/ -n1665 -vps -f')
//STEPLIB DD DSN=SSA.N3V100.SERVER.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPRNT2 DD SYSOUT=*
/*
```

To specify a list of populations to load when the SSA-NAME3 Server starts, use the `-p` option. The following example shows the usage of the `-p` option in the N3UP JCL:

```
//SSAN3UP JOB , 'N3UP', REGION=150M, NOTIFY=&SYSUID
//SSAN3SV EXEC PGM=SSAN3SV,
// PARM=('ENVAR("SSAPR=SSA.N3V100.PR")'),
// '/ -n1665 -pcanada,internat -vps -f')
//STEPLIB DD DSN=SSA.N3V100.SERVER.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPRNT2 DD SYSOUT=*
```

Informatica recommends that you use the `-p` option because it improves the performance of the first calls to SSA-NAME3. This option prevents concurrency issues when multiple user tasks connect to the server during the server startup process.

Note: Due to the 8-character limitation for the z/OS dataset member names, shorten the population names to 8 characters and remove the underscore (`_`) character if it is present. For example, you can shorten `international` to `internat` and `new_zealand` to `newzeala`.

Use the following guidelines when you customize the SSA-NAME3 Server JCLs:

- Select an appropriate port. In the preceding example, the port number that the SSA-NAME3 Server uses is 1665. The `SSAN3-CONNECT` call in the sample CICS program specifies this port number. If you change the port number in the JCL, change the port number in the sample program.

Note: Ensure that the same port number is not already defined in the sample CICS program.

- Change the `SSAPR=` parameter to match the dataset name where you install the standard populations. Omit the low-level qualifier, `DEFAULT`.

Note: You can use the `RPTSTG(ON)` parameter to generate a report on storage utilization of the SSA-NAME3 Server when you shut down the server. Informatica recommends that you use this parameter only to tune the z/OS operating system and remove it in a production environment because this parameter affects the performance of the server.

For example, `// PARM='RPTSTG(ON),ENVAR("SSAPR=SSA.N3V100.PR") / -n1665 -vps -f1'`

- Change the name of the `SERVER.LOAD` library to the name that you set.
- Use the `-vps` option on the `PARM` card to print the server progress messages. For example:

```
ssan3sv> Server launched: Wed Jun 4 03:23:18 2003
ssan3sv> Waiting for server to finish: Wed Jun 4 03:23:18 2003
ssan3sv> Server started: Wed Jun 4 03:23:18 2003
```

- Retain the `REGION` parameter on the `JOB` card. The `EXEC` card does not require the `REGION` parameter. Customize the N3DOWN JCL, and submit the job to cleanly shut down the SSA-NAME3 Server.

Note: Close all the open sessions of the server to cleanly shut down the server.

Transferring Custom Population Files

If it is required to transfer a Custom Population to the mainframe, then follow the procedure detailed here.

Note: The majority of users will probably use a Standard Population which is transferred using the procedure described in the Step 2 - Transfer the Datasets section in the Installing SSA-NAME3 chapter.

First, ensure that a PDS dataset exists to contain the Custom Population. It should have a name similar to `SSA.N3V100.PR.DEFAULT` and should have the following characteristics:

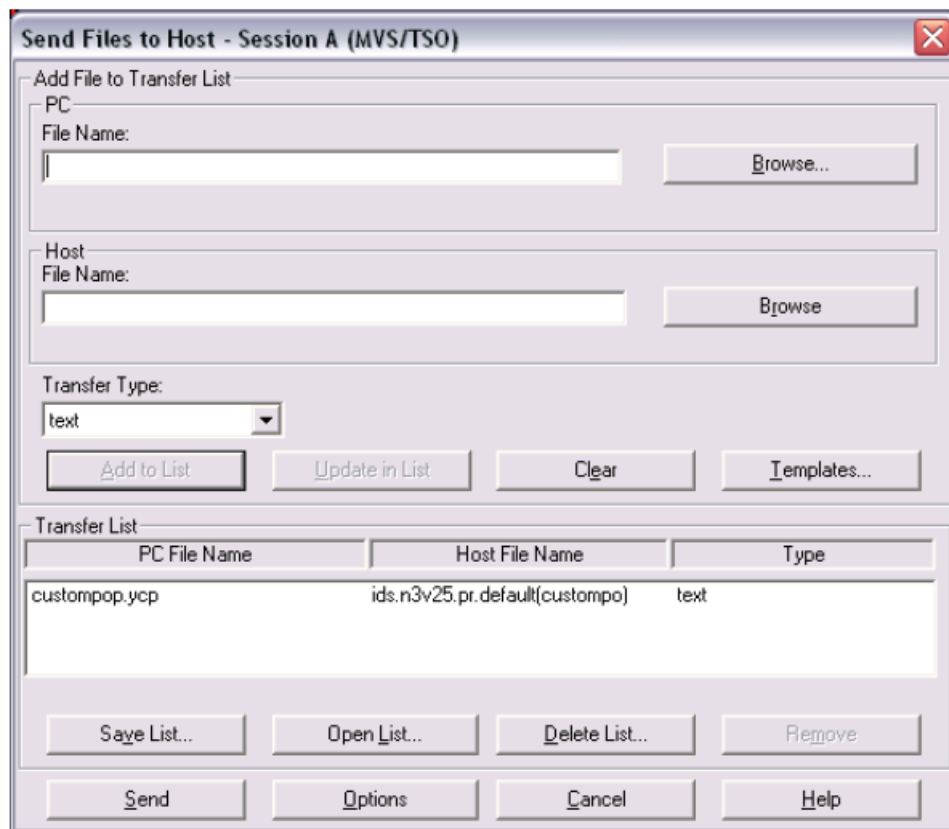
```
Organization . . . : PO
Record format . . . : VB
Record length . . . : 2184
```

```
Block size . . . . : 21840
Directory Blocks. . : 5
Primary Cylinders . : 10
Secondary Cylinders : 5
```

The Custom Population file must be transferred in Text (ASCII) mode. Also, the transfer can be performed directly into the `SSA.N3V100.PR.DEFAULT` dataset; no TSO Receive is required.

The member names in PDS datasets do not have extensions and have a name length restriction of 8 characters, so it may be necessary to give the Custom Population a new name. For example, a Custom Population called `australia.ysp` on the Windows computer might be copied to a PDS and given a member name `AUSTRALI`.

The following shows an `IND$FILE` transfer of an individual Population file from the Windows computer to the mainframe.



Note: Use of "Text" instead of "Binary" for the transfer. Also remove any RECFM, LRECL or SPACE options from the transfer screen.

Use the following example as a guide when transferring Custom Populations using FTP:

```
ftp> put australia.ysp 'nm3.n3v23.pr.default(australi)'
```

Testing SSA-NAME3

The following steps are needed to test SSA-NAME3 using one of the sample programs.

Testing the Batch Cobol Sample

The batch Cobol sample program is provided in `SSA.N3V100.SOURCE (COBSAMP)`. It can call `SSA-NAME3` either dynamically or statically. The sample provided shows the dynamic call.

Note: The Cobol sample is set up to use the USA Standard Population. This can be changed by editing the source.

Before compiling the program, check the `CONTROLS` parameter in the **SSAN3-OPEN** call. It specifies the name of the dataset where the Standard Populations are installed. By default it points to `SSAPR=SSA.N3V100.PR`. The low-level qualifier, example, `DEFAULT` is not specified. This must be modified to match the actual location where the Standard Populations were installed.

Compile the batch Cobol sample program (`COBSAMP`) using the example JCL supplied in `SSA.N3V100.SOURCE (COBCOMP)`.

The JCL must first be customized to your environment.

```
//SSAC001 JOB , 'COBCMP', NOTIFY=&SYSUID
//STEPS EXEC PROC=IGYWCL, LIBPRFX=CEE,
// PARM.COBOLE='RMODE (ANY), LIB',
// PARM.LKED='RMODE=ANY, AMODE=31'
//COBOL.SYSLIB DD DSN=SSA.N3V100.SOURCE, DISP=SHR
//COBOL.SYSIN DD DSN=SSA.N3V100.SOURCE (COBSAMP), DISP=SHR
//LKED.SYSLMOD DD DSN=YOUR.TEST.LOAD (TESTCOB), DISP=SHR
//LKED.SYSLIB DD DSN=CEE.SCEELKED, DISP=SHR
// DD DSN=SSA.N3V100.COBODYN.LOAD, DISP=SHR
//
```

To execute the batch Cobol program, run the JCL in `SSA.N3V100.SOURCE (COBRUN)` after customizing to your environment.

```
//SSAR001 JOB , 'COBRUN', NOTIFY=&SYSUID
//COBRUN1 EXEC PGM=TESTCOB, REGION=0M,
// PARM=' / '
// *PARM=' / POSIX (ON), RPTOPTS (ON) '
//STEPLIB DD DSN=SSA.N3V100.CORE.LOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSPRNT2 DD SYSOUT=*
//
```

Testing the Batch PL/I Sample

The batch PL/I sample program is provided in `SSA.N3V100.SOURCE (PLI2)`. PL/I can call `SSA-NAME3` either dynamically or statically. The sample provided shows the static call.

The member `SSA.N3V100.SOURCE (PLI1)` contains the JCL to compile, link and run the PL/I source module `PLI2`. Notice that for static calls the module `SSA.N3V100.CORE.LOAD (PLIOPEN)` must be linked to the PL/I program.

The JCL must be customized to your environment.

```
//SSAPLI1 JOB , 'TEST', NOTIFY=IBMUSER
//STEP1 EXEC PROC=IEL1CL
//STEPLIB DD DSN=IEL111.SIELCOMP, DISP=SHR
//SYSPUNCH DD SYSOUT=B
//SYSUT1 DD UNIT=SYSDA, SPACE=(1024, (60,60), , CONTIG)
//SYSPRINT DD SYSOUT=A
//PLI.SYSIN DD DSN=SSA.N3V100.SOURCE (PLI2), DISP=SHR
//LKED.SYSLMOD DD DSN=YOUR.TEST.LOAD (PLITEST), DISP=SHR
//LKED.SYSLIB DD DSN=CEE.SCEELKED, DISP=SHR
//SSALIB DD DSN=SSA.N3V100.CORE.LOAD, DISP=SHR
//LKED.SYSIN DD *
INCLUDE SSALIB (PLIOPEN)
NAME PLITEST (R)
/*
```

```

// *
// PLITEST EXEC PGM=PLITEST,REGION=4M,
// PARM=' / POSIX(OFF) '
// STEPLIB DD DSN=SSA.N3V100.PR.DEFAULT.LOAD,DISP=SHR
// DD DSN=SSA.N3V100.CORE.LOAD,DISP=SHR
// * DD DSN=YOUR.TEST.LOAD,DISP=SHR
// * DD DSN=CEE.SCEERUN,DISP=SHR
// SYSPRINT DD SYSOUT=*
// SYSPRNT2 DD SYSOUT=*
// *

```

Testing the CICS Cobol Sample

The CICS Cobol sample program is provided as member `SSA.N3V100.SOURCE(CICSCOB)`. It will demonstrate how to call the SSA-NAME3 Server.

Step 1 - Compile the Sample Program

Customize the JCL `SSA.N3V100.SOURCE(CICSCOMP)` which will be used to compile `CICSCOB` and link-edit it as `SSATST1`. Run the job `CICSCOMP` and then copy the module `SSA.N3V100.CORE.LOAD(SSATST1)` into a CICS Load library.

Step 2 - Create a CICS Transaction to run the Sample Program

Create a CICS transaction to run the program `SSATST1`. See the transaction definition `SSA3` example in the Appendix.

Step 3 - Ensure CICS Socket Support is Enabled

The minimum requirements section talked about the need for CICS TCP/IP Socket Interface. Before testing the CICS sample program, CICS TCP/IP Socket Interface must be enabled. This would typically be done by a systems programmer through the `CICS EZAO` transaction.

Step 4 - Start the SSA-NAME3 Server

Start the SSA-NAME3 Server from `SSA.N3V100.SOURCE(N3UP)`. This should already have been customized to suit your environment.

Step 5 - Run the Transaction

Run the transaction and check the results. As supplied, the Cobol program writes results to the MSG Queue `CSMT`.

The results should look something like:

```

SSA-CONNECT
Socket : 000000000
Response Code : 0
Message :
SSA-OPEN
SessionID : 016777216
Socket : 000000000
Controls : SSAPR=SSA.N3V100.PR
Response Code : 0
Message :
SSA-GET-KEYS
SessionID : 016777216
Socket : 000000000
Controls : FIELD=ADDRESS_PART1 KEY LEVEL=STANDARD
Input Record : *ADDRESS_PART1*N 9391 GREEN OAKS TR CT HWY 53 LT 5***
Response Code : 0
Message :
Count : 000000068
USOBZTB>
USZYDTZY
USGBZPAUS>
RZPAUSGZZPAUSZXDKZY

```

USZXC\$ZY
ZZE\$?*ZY
ZZE\$BZMI
ZZE\$/*OU
ZZE\$?AZY
ZZE\$*?ZY
ZZE\$/BZY
ZZE\$BZEI
ZZEG?*ZY
ZZEGBZMI
ZZEG/*OU
ZZEG?AZY
ZZEG*?ZY
ZZEG/BZY
ZZEGBZERRFW?*
ZY
RRFWBZMI
RRFW/*OU
RRFW?AZY
RRFW*?ZY
RRFWBZEI
RRFWBZE-
MOOBZTBA
MOZYDTOU
MOGANJYQ
MOOVZPA
MOGZZPA
MOZXDKZY
MOZXC\$ZY
QNOBZTBA
QNZYDTOU
QNOVZPA
QN>RZPA
QNGZZPA
QNZXDKZY
QNZXC\$ZY
ZZMIH*OU
ZZMIF*OU
ZZMIHAZY
ZZMID?ZY
ZZMIFBZY
ZZMIJZEI
ZZMIJZE-
URROBZMI
URRO/*OU
URRO?AZY
URRO*?ZY
URRO/BZY
URROBZEI
URROBZE-
URRRN*OU
URRRPAZY
URRRL?ZY
URRRNBZY
URRRRZEI
URRRRZE-
ZZEHH*ZY
ZZEHJZMI
ZZEHF*OU
ZZEHHAZY
ZZEHD?ZY
ZZEHFBZY
SSA-CLOSE
SessionID : 00000000J
Socket : 000000000
Controls : FIELD=ADDRESS_PART1 KEY_LEVEL=STANDARD
Response Code : 0
Message :
SSA-DISCONNECT
Socket : 000000000

Response Code : 0
Message :

Appendix

This section describes information that you need to know during the installation of SSA-NAME3.

CICS Notes

Calling SSA-NAME3 from CICS Applications

There are several ways that CICS applications can call SSA-NAME3.

Database Stored Procedures

If the database being used supports multithreaded stored procedures, the applications that call SSA-NAME3 may be implemented as database stored procedures that dynamically call the SSA-NAME3 DLL.

SSA-NAME3 Server

The CICS application can call the SSA-NAME3 Server, a process which is started in a separate region. CICS applications communicate with the SSA-NAME3 Server through a SSA-NAME3 stub program which uses sockets and TCP/IP. The additional requirements for CICS applications calling the SSA-NAME3 Server are therefore:

1. CICS TCP/IP Socket Interface support must be installed, configured and enabled.
2. Language Environment (LE) must be enabled in CICS.

SSA-NAME3 DLL

The CICS application can call the SSA-NAME3 DLL locally, within the CICS region. Using this method, it is the user application's responsibility to serialize the Open and Close calls to SSA-NAME3. The recommended approach for doing this is through the ENQ/DEQ commands using a fixed name ENQ as illustrated below:

```
01 SSAN3SID PIC X(8) VALUE "SSAN3SID".  
  
EXEC CICS ENQ RESOURCE(SSAN3SID) LENGTH(8) END-EXEC  
CALL SSAN3-OPEN .... (or SSAN3-CLOSE)  
EXEC CICS DEQ RESOURCE(SSAN3SID) LENGTH(8) END-EXEC
```

Using this method, LE must also be enabled in CICS.

If multiple CICS regions are involved, ENQ/DEQ can be 'externalized' across LPARs / CICS regions and managed by Global Resource Serialization (GRS) using appropriate ENQ MODEL resource definitions.

Copybooks

The following Copybooks are supplied in `SSA.N3V100.SOURCE` and are provided for CICS applications that will call the SSA-NAME3 Server.

SSAN3WS contains working storage definitions

SSAN3PD contains procedure division set-up for use of `ssan3fn`

SSAN3P2 contains procedure division cleanup (releases allocated resources)

Possible Problems

The possible problems and workarounds are as shown in the table below:

Possible Problem	Possible Cause	Action
CICS abend AEY9	May mean that the CICS Socket Interface has not been started.	Use EZAO START CICS to start it.
CICS Cobol call (SSAN3-CONNECT) fails with an error: SSACONN: RSP: 1 Connection to ez server 'localhost:1665' failed -221	The mainframe TCPIP set-up does not have a Name Server or Host file set-up correctly.	InTCPIP.HOSTS.LOCAL the host that is to be used should be listed or a valid Name server specified using the NSINTERADDR statement in TCPIP.TCPIP.DATA. If the host is added to TCPIP.HOSTS.LOCAL then the makesite command should be rerun. For example: makesite HLQ=TCPIP.

Program Definitions

```

VIEW PROGRAM(CSSAIOK) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS
CEDA View PROGram( CSSAIOK )
Program      : CSSAIOK
Group       : IDSGROUP
DEscription  :
Language    : C
REload      : No
RESident    : No
USAge       : Normal
USElpacopy   : No
Status      : Enabled
RSl         : 00
CEdf        : Yes
DAtalocation : Any
EXECKey     : User
CONcurrency  : Quasirent
REMOTE ATTRIBUTES
DYNAMIC      : No
+ REMOTESystem : SYSID=CICS APPLID=CICS
VIEW PROGRAM(SSAN3CC) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS CICS RELEASE = 0610
CEDA View PROGram( SSAN3CC )
Program      : SSAN3CC
Group       : IDSGROUP
DEscription  :
Language    : C
REload      : No
RESident    : No
USAge       : Normal
USElpacopy   : No
Status      : Enabled
RSl         : 00
CEdf        : Yes
DAtalocation : Any
EXECKey     : User
CONcurrency  : Threadsafe
REMOTE ATTRIBUTES
DYNAMIC      : No
+ REMOTESystem : SYSID=CICS APPLID=CICS
VIEW PROGRAM(SSAOPEN) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS CICS RELEASE = 0610
CEDA View PROGram( SSAOPEN )

```

CICS RELEASE = 0610

CObol | Assembler | Le370 | C | Pl1
 No | Yes
 No | Yes
 Normal | Transient
 No | Yes
 Enabled | Disabled
 0-24 | Public
 Yes | No
 Below | Any
 User | Cics
 Quasirent | Threadsafe

No | Yes

CObol | Assembler | Le370 | C | Pl1
 No | Yes
 No | Yes
 Normal | Transient
 No | Yes
 Enabled | Disabled
 0-24 | Public
 Yes | No
 Below | Any
 User | Cics
 Quasirent | Threadsafe

No | Yes

```

PROGRAM      : SSAOPEN
Group        : IDSGROUP
Description   :
Language     : C
RELoad       : No
RESident     : No
USAge        : Normal
USElpacopy   : No
Status       : Enabled
RS1          : 00
CEdf         : Yes
Datalocation : Any
EXECKey      : User
Concurrency  : Threadsafe
REMOTE ATTRIBUTES
Dynamic      : No
+ REMOTESystem : SYSID=CICS APPLID=CICS
VIEW PROGRAM(SSAN3FN) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS CICS RELEASE = 0610
CEDA View PROGRAM( SSAN3FN )
PROGRAM      : SSAN3FN
Group        : IDSGROUP
Description   :
Language     : C
RELoad       : No
RESident     : No
USAge        : Normal
USElpacopy   : No
Status       : Enabled
RS1          : 00
CEdf         : Yes
Datalocation : Any
EXECKey      : User
Concurrency  : Threadsafe
REMOTE ATTRIBUTES
Dynamic      : No
+ REMOTESystem : SYSID=CICS APPLID=CICS
VIEW PROGRAM(SSATST1) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS CICS RELEASE = 0610
CEDA View PROGRAM( SSATST1 )
PROGRAM      : SSATST1
Group        : IDSGROUP
Description   :
Language     : CObol
RELoad       : No
RESident     : No
USAge        : Normal
USElpacopy   : No
Status       : Enabled
RS1          : 00
CEdf         : Yes
Datalocation : Any
EXECKey      : User
Concurrency  : Threadsafe
REMOTE ATTRIBUTES
Dynamic      : No
+ REMOTESystem : SYSID=CICS APPLID=CICS
VIEW PROGRAM(SSACLOSE) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS CICS RELEASE = 0610
CEDA View PROGRAM( SSACLOSE )
PROGRAM      : SSACLOSE
Group        : IDSGROUP
Description   :
Language     : C
RELoad       : No

```

CObol | Assembler | Le370 | C | Pl1
No | Yes
No | Yes
Normal | Transient
No | Yes
Enabled | Disabled
0-24 | Public
Yes | No
Below | Any
User | Cics
Quasirent | Threadsafe
No | Yes

CObol | Assembler | Le370 | C | Pl1
No | Yes
No | Yes
Normal | Transient
No | Yes
Enabled | Disabled
0-24 | Public
Yes | No
Below | Any
User | Cics
Quasirent | Threadsafe
No | Yes

CObol | Assembler | Le370 | C | Pl1
No | Yes
No | Yes
Normal | Transient
No | Yes
Enabled | Disabled
0-24 | Public
Yes | No
Below | Any
User | Cics
Quasirent | Threadsafe
No | Yes

Additional entries for Natural

```

VIEW PROGRAM(SSACLOSE) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS CICS RELEASE = 0610
CEDA View PROGRAM( SSACLOSE )
PROGRAM      : SSACLOSE
Group        : IDSGROUP
Description   :
Language     : C
RELoad       : No

```

CObol | Assembler | Le370 | C | Pl1
No | Yes

RESident	: No	No Yes
USAge	: Normal	Normal Transient
USElpacopy	: No	No Yes
Status	: Enabled	Enabled Disabled
RS1	: 00	0-24 Public
CEdf	: Yes	Yes No
DAtalocation	: Any	Below Any
EXECKey	: User	User Cics
Concurrency	: Threadsafe	Quasirent Threadsafe
REMOTE ATTRIBUTES		
DYnamic	: No	No Yes

VIEW PROGRAM(GETKEYS) GROUP(IDSGROUP)
 OBJECT CHARACTERISTICS CICS RELEASE = 0610

CEDA View PROGram(GETKEYS)		
PROgram	: GETKEYS	
Group	: IDSGROUP	
DEscription	:	
Language	: C	CObol Assembler Le370 C Pli
RELoad	: No	No Yes
RESident	: No	No Yes
USAge	: Normal	Normal Transient
USElpacopy	: No	No Yes
Status	: Enabled	Enabled Disabled
RS1	: 00	0-24 Public
CEdf	: Yes	Yes No
DAtalocation	: Any	Below Any
EXECKey	: User	User Cics
Concurrency	: Threadsafe	Quasirent Threadsafe
REMOTE ATTRIBUTES		
DYnamic	: No	No Yes

VIEW PROGRAM(GETRNGES) GROUP(IDSGROUP)
 OBJECT CHARACTERISTICS CICS RELEASE =
 0610

CEDA View PROGram(GETRNGES)		
PROgram	: GETRNGES	
Group	: IDSGROUP	
DEscription	:	
Language	: C	CObol Assembler Le370 C Pli
RELoad	: No	No Yes
RESident	: No	No Yes
USAge	: Normal	Normal Transient
USElpacopy	: No	No Yes
Status	: Enabled	Enabled Disabled
RS1	: 00	0-24 Public
CEdf	: Yes	Yes No
DAtalocation	: Any	Below Any
EXECKey	: User	User Cics
Concurrency	: Threadsafe	Quasirent Threadsafe
REMOTE ATTRIBUTES		
DYnamic	: No	No Yes

VIEW PROGRAM(SSAMATCH) GROUP(IDSGROUP)
 OBJECT CHARACTERISTICS CICS RELEASE = 0610

CEDA View PROGram(SSAMATCH)		
PROgram	: SSAMATCH	
Group	: IDSGROUP	
DEscription	:	
Language	: C	CObol Assembler Le370 C Pli
RELoad	: No	No Yes
RESident	: No	No Yes
USAge	: Normal	Normal Transient
USElpacopy	: No	No Yes
Status	: Enabled	Enabled Disabled
RS1	: 00	0-24 Public
CEdf	: Yes	Yes No
DAtalocation	: Any	Below Any
EXECKey	: User	User Cics
Concurrency	: Threadsafe	Quasirent Threadsafe

```

REMOTE ATTRIBUTES
Dynamic      : No
No | Yes

```

Optional entry for Natural if using the SSAINFO api

```

VIEW PROGRAM(SSAINFO) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS
CEDA View Program(SSAINFO)
PROGRAM      : SSAINFO
Group       : IDSGROUP
DEscription  :
Language    : C
RELoad      : No
RESident    : No
USAge       : Normal
USElpacopy   : No
Status      : Enabled
RS1         : 00
CEdf        : Yes
DAtalocation : Any
EXEckey     : User
CONcurrency  : Threadsafe
REMOTE ATTRIBUTES
Dynamic      : No
No | Yes
CICS RELEASE = 0610
CObol | Assembler | Le370 | C | PlI
No | Yes
No | Yes
Normal | Transient
No | Yes
Enabled | Disabled
0-24 | Public
Yes | No
Below | Any
User | Cics
Quasirent | Threadsafe

```

Optional entries for Natural if using the SSA-NAME3 Server

```

VIEW PROGRAM(SSACONN) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS
CEDA View Program(SSACONN)
PROGRAM      : SSACONN
Group       : IDSGROUP
DEscription  :
Language    : C
RELoad      : No
RESident    : No
USAge       : Normal
USElpacopy   : No
Status      : Enabled
RS1         : 00
CEdf        : Yes
DAtalocation : Any
EXEckey     : User
CONcurrency  : Threadsafe
REMOTE ATTRIBUTES
Dynamic      : No
No | Yes
CICS RELEASE = 0610
CObol | Assembler | Le370 | C | PlI
No | Yes
No | Yes
Normal | Transient
No | Yes
Enabled | Disabled
0-24 | Public
Yes | No
Below | Any
User | Cics
Quasirent | Threadsafe

VIEW PROGRAM(SSADISC) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS
CEDA View Program(SSADISC)
PROGRAM      : SSADISC
Group       : IDSGROUP
DEscription  :
Language    : C
RELoad      : No
RESident    : No
USAge       : Normal
USElpacopy   : No
Status      : Enabled
RS1         : 00
CEdf        : Yes
DAtalocation : Any
EXEckey     : User
CONcurrency  : Threadsafe
REMOTE ATTRIBUTES
Dynamic      : No
No | Yes
CICS RELEASE = 0610
CObol | Assembler | Le370 | C | PlI
No | Yes
No | Yes
Normal | Transient
No | Yes
Enabled | Disabled
0-24 | Public
Yes | No
Below | Any
User | Cics
Quasirent | Threadsafe

```


Transaction Definition

```
VIEW TRANS(SSA3) GROUP(IDSGROUP)
OBJECT CHARACTERISTICS
CEDA View TRANSaction( SSA3 )
TRANSaction      : SSA3
Group            : IDSGROUP
DEscription      :
PROgram          : SSATST1
TWAsize          : 00000 0-32767
PROfile          : DFHCICST
PARTitionset     :
STatus           : Enabled           Enabled | Disabled
PRIMedsize       : 00000             0-65520
TASKDATAloc      : Any               Below | Any
TASKDATAkey      : User              User | Cics
STOrageclear     : No                No | Yes
RUNaway          : System            System | 0 | 500-2700000
SHutdown         : Disabled          Disabled | Enabled
ISolate          : Yes               Yes | No
BrexIt           :
+ REMOTE ATTRIBUTES
SYSID=CICS APPLID=CICS
```

Other Cobol Notes

Copy books are supplied in `SSA.N3V100.SOURCE`. Look at the supplied sample source code for examples on the use of these copybooks.

To statically link `SSA-NAME3` to a Cobol program, link the Cobol program to the module `SSA.N3V100.CORE.LOAD(SSALOAD)`.

Other PL/I Notes

The example program `PLI2` demonstrates how to call `SSA-NAME3`.

There are strict requirements for the parameters for a PL/I call to the `SSA-NAME3` API. The parameters must be defined as follows (these examples are for an `SSAOPEN` call):

```
DCL ADDR BUILTIN;
DCL J FIXED BIN(31,0);
DCL SSAOPEN EXTERNAL ENTRY RETURNS(FIXED BIN(31,0));
DCL SSACLOS EXTERNAL ENTRY RETURNS(FIXED BIN(31,0));
DCL GETKEYS EXTERNAL ENTRY RETURNS(FIXED BIN(31,0));
DCL GETRNGS EXTERNAL ENTRY RETURNS(FIXED BIN(31,0));
DCL SSAINFO EXTERNAL ENTRY RETURNS(FIXED BIN(31,0));
DCL SSAMTCH EXTERNAL ENTRY RETURNS(FIXED BIN(31,0));
DCL SOCKH FIXED BIN(31,0) INIT(-1);
DCL SESSION FIXED BIN(31,0) INIT(-1);
DCL SYSTEM PTR;
DCL POPULATION PTR;
DCL CONTROLS PTR;
DCL SSARSP PTR;
DCL SSAMSG PTR;
DCL P1 CHAR(32) INIT(' ');
DCL P2 CHAR(32) INIT(' ');
DCL P3 CHAR(1924) INIT(' ');
DCL P4 CHAR(2) INIT('99');
DCL P5 CHAR(1024) INIT(' ')
```

The CHAR parameters are setup as follows: The System, Population and Controls must be null terminated using `"||'00'x"`.

```
SYSTEM = ADDR(P1);
POPULATION = ADDR(P2);
```

```

CONTROLS = ADDR(P3);
SSARSP = ADDR(P4);
SSAMSG = ADDR(P5);
P1 = 'default' || '00'x;
P2 = 'usa' || '00'x;
P3 = 'FIELD=PERSON_NAME' || '00'x

```

Finally the call is done as follows:

```
J = SSAOPEN( ADDR(SOCKH), ADDR(SESSION), ADDR(SYSTEM), ADDR(POPULATION), ADDR(CONTROLS),
```

Do not leave out any of these steps otherwise the call will fail with an abend or simply not return anything. Make sure SOCKH and SESSION are set to -1 before the call to SSAOPEN.

The entry point names differ slightly from the Cobol interface as PL/I is limited to 7 character external names.

The entry point names for the static call are:

```

SSADISC
SSACONN
SSAOPEN
SSACLOS
GETKEYS
SSAINFO
SSAMTCH
GETRNGS

```

Alternatively the SSA.N3V100.PLIDYN.LOAD library contains entry points that can be called dynamically.

Natural

The Natural option LE370=ON is required.

To install SSA-NAME3 in a batch NATURAL environment, add SSA.N3V100.NAT.LOAD in the STEPLIB before SSA.N3V100.CORE.LOAD.

To install SSA-NAME3 in a CICS NATURAL environment, follow the CICS install instructions to use either population ysg or DLL file. In addition, add PPT entries for each of the modules in the SSA.N3V100.NAT.LOAD dataset and copy these modules to a CICS DFHRPL dataset. Sample PPT entries are listed earlier in this Appendix.

Note: Be sure to use the SSAOPEN from SSA.N3V100.NAT.LOAD instead of IDS.N3V27.CORE.LOAD.

SSAOPEN, SSACLOSE, GETKEYS, GETRNGES, SSAMATCH and, if the apis are used in the application, **SSAINFO,SSACONN** and **SSADISC** should not be defined to NATURAL using either CSTATIC or RCA.

Contents of Release Datasets

```

DSLST
SSA.N3V100.SERVER.LOAD

```

Row 00001 of 00013

```

Command
===>

```

Scroll ==> PAGE

Name Prompt Alias-of Size TTR AC AM RM

```

_____ PLIOPEN
_____ SSAIOK
_____ SSALSN
_____ SSAN3CA
_____ SSAN3CC
_____ SSAN3CL

```

```
----- SSAN3CP
----- SSAN3SV
----- SSAN3TB
----- SSAN3V2
----- SSAOPEN
----- SSASHUT
----- TESTIT
----- VERSION
```

BROWSE
SSA.N3V100.CORE.LOAD

Row 00001 of 00015

Command
===>

Scroll ===> PAGE

Name Prompt Alias-of Size TTR AC AM RM

```
----- CSSAIOK
----- PLIOPEN
----- PLIOPENC
----- PLITEST
----- SSAIOK
----- SSALSN
----- SSAN3CC
----- SSAN3CL
----- SSAN3FN
----- SSAN3TB
----- SSAN3V2
----- SSAOPEN
----- SSAOPENC
----- SSATST1
----- TEMPNAM0
----- TESTCOB
```

DSLIST
SSA.N3V100.SOURCE

Row 00001 of

00013
Command
===>

Scroll ===> PAGE

Name Prompt Size Created Changed ID

```
----- CICSCOB
----- CICSCOMP
----- COBCOMP
----- COBRUN
----- COBSAMP
----- N3DOWN
----- N3UP
----- PLI1
----- PLI2
----- SSAN3PD
----- SSAN3P2
----- SSAN3WS
----- VERSION
```

BROWSE
SSA.N3V100.PLIDYN.LOAD

Row 00001 of 00008

Command
===>

Scroll ===> PAGE

Name Prompt Alias-of Size TTR AC AM RM

-----	GETKEYS
-----	GETRNGS
-----	SSACLOS
-----	SSACONN
-----	SSADISC
-----	SSAINFO
-----	SSAMTCH
-----	SSAOPEN

Populations available

AML
 ARABIC
 ARABICR
 ARGENTIN
 AUSTRALI
 BELGIUM
 BRASIL
 CANADA
 CHILE
 CHINESER
 COLOMBIA
 CZECH
 DENMARK
 ESTONIA
 FINLAND
 FRANCE
 GAELIC
 GERMANY
 GREEK
 HEBREW
 HKR
 HUNGARY
 INDIA
 INDONESIA
 INTERNAT
 IRELAND
 ITALY
 JAPANR
 KAZAKHST
 KOREANR
 LUXEMBOU
 MALAYSIA
 MEXICO
 NETHERLA
 NEWZEALA
 NORWAY
 OFAC
 PERU
 PHILIPPI
 POLAND
 PORTUGAL
 PUERTORI
 RUSSIA
 SINGAPOR
 SOUTHAFR
 SPAIN
 SWEDEN
 SWITZERL
 TAIWANR
 THAIR
 TURKEY
 UK
 USA
 VIETNAMR

Populations requiring CJK license

ARABICM
CHINESEI
CHINESES
CHINESET
HK
KOREAN
JAPANI
JAPAN
TAIWAN
THAI
VIETNAM

INDEX

C

CEDA [21](#)
CICS [16](#), [17](#), [21](#), [26](#)
CICS EZAO [26](#)
CICS Socket Support [26](#)
CJK-SUPPORT [15](#), [18](#)
Cobol [25](#)
CONTROLS parameter [25](#)
Custom Population [23](#)

D

Developer's Workbench [10](#), [12](#), [15](#)
DLL/Shared Library [15](#)

F

files
 dll [18](#)
 ysp [18](#)
FTP Transfer [18](#)

I

IND\$FILE [18](#)
Installation Sequence [10](#)

J

Java GUI [10](#)
JCL [16](#), [17](#), [22](#), [25](#)

L

Language Environment [16](#)
license server [10](#)
Load datasets [18](#)
Load modules [18](#)

M

MSG Queue [26](#)

N

N3V100.DLL [18](#)

P

Parameter
 REGION [22](#)
 SSAPR= [22](#)
PATH variable [15](#)
PDS dataset [23](#)
PL/I [17](#), [25](#)
Population DLL [21](#)

R

RACF [16](#)
REGION parameter [22](#)

S

server software [10](#)
shared library [15](#)
Source dataset [18](#)
SSA-NAME3 Callable Routine [10](#), [15](#)
SSA-NAME3 mainframe [16](#)
SSA-NAME3 Server [16](#)
SSAN3-OPEN [25](#)
Standard Population [23](#)
Standard Populations [14](#)
stub program [16](#)

T

TCP/IP [16](#)
TCP/IP connection [11](#)
TSO [18](#), [23](#)
TSO RECEIVE [18](#)
TSO TRANSMIT [18](#)

W

Workbench [10](#)