

SOLA™

SOLA
Version 6.1

Installation Guide
SOLA CICS Container
SOLA IMS Container
SOLA Development Studio
SOLA Resource Manager

Revision Date: June 26, 2013

Trademarks

SOA Software and the SOA Software logo are either trademarks or registered trademarks of SOA Software, Inc. Other product names, logos, designs, titles, words or phrases mentioned within this guide may be trademarks, service marks or trade names of SOA Software, Inc. or other third parties and may be registered in the U.S. or other jurisdictions.

Copyright

©2001-2010 SOA Software, Inc. All rights reserved. No material in this manual may be copied, reproduced, republished, uploaded, posted, transmitted, distributed or converted to any electronic or machine-readable form in whole or in part without prior written approval from SOA Software, Inc.

Table of Contents

INSTALLATION GUIDE	I
PREFACE	VII
Before You Begin	vii
Installation Components	ix
Overview of SOLA Installation Process	x
In This Guide	x
System Requirements	xii
Customer Support	xiv
CHAPTER 1: INSTALLING THE SOLA RUN-TIME ON A Z/SERIES MAINFRAME	1
Installation Requirements and Considerations	1
Driving System Requirements	1
Target System Requirements	3
Installation Instructions	7
Overview of SOLA for z/OS installation	8
Unzip the distribution files	11
Transmit the distribution files to the mainframe	13
Create the install library	17
Customize WRKSHEET for your environment	18
Receive the distribution files	22
Customize the sample installation JCL	24
Use SMP/E to install sola for z/OS	26
CHAPTER 2: CUSTOMIZING SOLA CICS CONTAINER ON A Z/SERIES MAINFRAME (SOLA600)	31
Customizing the SOLA installation	31
Customizing the <liblq>.SAMPLIB	31
Creating the Database with DDLJCL	32
Binding the Plan and Packages with BINDJCL	33
Creating the <vsamUMT> and <vsamCMT> files with IDCAMJCL	33
Define the CICS CSD entries with CSDJCL	33
Define the CICS Analyzer to use with SOLA (Optional)	34
TCPIPS Definition	35
RACF Authorizing the SOLA Transactions (optional)	35
Defining IMS components (optional)	35
Setting up the IMS Message Processing Region (optional)	36
Populating data for sample application (Recommended)	36
Migration to SOLA6.1 from SOLA5.1 (Optional)	36
CHAPTER 3: CUSTOMIZING SOLA IMS CONTAINER ON A Z/SERIES MAINFRAME (SOLS600)	39
Customizing the SOLA installation	39
Customizing the <liblq>.SAMPLIB	39

Creating the Database with DDLJCL	40
Binding the Plan and Packages with BINDJCL	40
RACF Authorization of SOLA IMS Container	41
APF Authorize the SOLA Load Library (Recommended)	41
Defining IMS components	41
Setting up the IMS Message Processing Region (optional)	41
Configuring the SOLA IMS Container	42
CHAPTER 4: INSTALLING THE SOLA DEVELOPMENT STUDIO	47
Overview	47
Installing the SOLA Development Studio in WebSphere	48
Installing the SOLA Development Studio in WebLogic	58
Installing the SOLA Development Studio in Tomcat	65
CHAPTER 5: CUSTOMIZING SOLA	71
Development Studio Customization worksheet	72

Table of Figures

Figure 1 Driving System Software Requirements	1
Figure 2 Mandatory Requisites	3
Figure 3 Functional Requisites.....	4
Figure 4 Total DASD space required by SOLA for z/OS	6
Figure 5: WebSphere Application Server -Logon page	48
Figure 6: WebSphere Application Server -Integrated Solutions Console	49
Figure 7: WebSphere Application Server -Install new Application	49
Figure 8: WebSphere Application Server -Select the sola.ear file	50
Figure 9: WebSphere Application Server -Upload the sola.ear file	50
Figure 10: WebSphere Application Server -Select Installation Options	51
Figure 11: WebSphere Application Server -Map modules to servers	51
Figure 12: WebSphere Application Server -Map virtual hosts for web modules.....	52
Figure 13: WebSphere Application Server -Summary	52
Figure 14: WebSphere Application Server -Save	53
Figure 15: WebSphere Application Server -Configure the class loading policy	53
Figure 16: WebSphere Application Server -Class loading	54
Figure 17: WebSphere Application Server -General Properties	54
Figure 18: WebSphere Application Server -Configuring the SOLARoot	55
Figure 19: WebSphere Application Server -Server Infrastructure	55
Figure 20: WebSphere Application Server -Process Definition	56
Figure 21: WebSphere Application Server -JVM	56
Figure 22: WebSphere Application Server -JVM Properties	57
Figure 23: WebSphere Application Server -JVM Properties	57
Figure 24: WebLogic -Start-up	58
Figure 25: WebLogic Server -Administration Console.....	59
Figure 26: WebLogic -Server Home.....	59
Figure 27: WebLogic -Deploy a new application	60
Figure 28: WebLogic -Install or Update an Application	60
Figure 29: WebLogic -Choose file.....	61
Figure-30: WebLogic -File selected.....	61
Figure 31: WebLogic -Deploy an Application.....	62
Figure 32: WebLogic -Deploy	62
Figure 33: WebLogic -Successful Deployment.....	63
Figure 34: WebLogic -Start WebLogic Script.....	63
Figure 35: WebLogic -Start-up	64
Figure 36: Tomcat -war file copy	65
Figure 37: Tomcat -Edit server.xml	66
Figure 38: Tomcat -server.xml	66
Figure 39: Tomcat -Edit catalina.bat	67
Figure 40: Tomcat -catalina.bat.....	67
Figure 41: Tomcat -Start Tomcat.....	68
Figure 42: Tomcat -Command Prompt Window	68
Figure 43: Login with the installation ID	75
Figure 44: Enter the product ID	76
Figure 45: Read and agree to the license agreement	76
Figure 46: Accept the license agreement	77
Figure 47: Debugging XML	78
Figure 48: EndPoint Key Value pairs	79
Figure 49: endpoints changed.....	81

Figure 50: Selecting /Jobcard.txt.....	82
Figure 51: Modifying /Jobcard.txt	82
Figure 52: Creating Environments	83
Figure 53: Adding the SOLA Administrator.....	83
Figure 54: Creating a Container group and Container	84
Figure 55: Choosing the environment	84
Figure 56: Login with the SOLA Administrator Id	85
Figure 57: Create Container Group	86
Figure 58: Create Container Group (IMS version)	87
Figure 59: Container Group created successfully	88
Figure 60: Create a new container.....	88
Figure 61: Create a new container.....	89
Figure 62: New Container created	90

Preface

This installation guide provides instructions for installing, configuring, and running the SOLA Development Studio on a J2EE platform (WebSphere, WebLogic or Tomcat), and installing the SOLA run-time on z/OS using SMP/E. The first half of this manual is devoted to the SMP/E installation of the run-time and the second half addresses the Development Studio installation.

Note: This manual documents the installation of multiple SOLA products. To differentiate between the two mainframe products, SOLA CICS Container (FMID: SOLA600) and SOLA IMS Container (FMID: SOLS600), each product is referenced by its FMID, so sections on the installation of SOLA CICS Container are identified SOLA600, while sections on the installation of SOLA IMS Container are identified SOLS600.

BEFORE YOU BEGIN

The SOLA 6.1 installation process includes the following mandatory pre-installation steps for *new* and *upgrade* installations:

SOLA Pre-installation Requirements

SOLA Installation Scenario	Pre-installation Requirement
SOLA Server	<ul style="list-style-type: none"> ○ J2EE Application Server: Before installing the SOLA Development Studio into your J2EE application server you must first install the application server. SOLA supports installation of the SOLA Development Studio in WebSphere, WebLogic or Tomcat. Next you will need to create an installation directory in the file system of J2EE server and copy the installation EAR or WAR file(whichever is appropriate) from the installation package into this directory. Note: SOLA requires a JRE (Java Runtime Environment) of 1.5 or above.
Client Web Browser	<ul style="list-style-type: none"> ○ Cookies must be enabled in your browser settings in order to use the SOLA Development Studio
z/OS Server	<p>You will need the following systems and/or subsystems installed and running on your destination machine:</p> <ul style="list-style-type: none"> ○ For SOLA CICS Container (SOLA600): <ul style="list-style-type: none"> ○ G4 processor or higher for encryption (if required) ○ DB2 version 7 or greater ○ MVS/ESA 430 or higher ○ CICS TS 1.31 or greater ○ CICS CWS ○ CICS Sockets if outbound support and/or IMS is/are required ○ Bridge 3270 or Linkable Bridge for 3270 support ○ MQ if MQ transport is required

	<ul style="list-style-type: none"> ○ DB2 Version 8 or higher for UDDI inquiry & Dashboard functionality ○ MVS FTP server must be running for the development environment ○ Integrated Cryptographic Service Facility (ICSF) for XML Encryption, XML Signature and outbound SSL (if required) ○ Optional: IMS version 9 or greater, for access to IMS/TM transactions. ○ Optional: IMS Connect version 8 or greater and/or OTMA/CI, for access to IMS/TM transactions. ○ For SOLA IMS Container (SOLS600): <ul style="list-style-type: none"> ○ G4 processor or higher for encryption (if required) ○ DB2 version 7 or greater ○ MVS/ESA 430 or higher ○ DB2 Version 8 or higher for UDDI inquiry & Dashboard functionality ○ MVS FTP server must be running for the development environment ○ Integrated Cryptographic Service Facility (ICSF) for XML Encryption, XML Signature and outbound SSL (if required) ○ IMS version 9 or greater, for access to IMS/TM transactions. ○ IMS Connect version 8 or greater and/or OTMA/CI, for access to IMS/TM transactions.
--	---

INSTALLATION COMPONENTS

A complete SOLA installation includes a variety of different components to accommodate your SOA creation, publishing, management and monitoring needs.

The following table lists the SOLA Installation Components:

Installation Components

SOLA Component	Description
SOLA CICS Container (SOLA600)	The z/Series mainframe CICS run-time component. This component can be used as a backend for the SOLA Development Studio and as the SOAP runtime engine that provides SOAP capability for the z/Series mainframe.
SOLA IMS Container (SOLS600)	The z/Series mainframe Started Task Address Space run-time component. This component can be used as a backend for the SOLA Development Studio and as the SOAP runtime engine that provides SOAP capability to IMS transactions that execute on the z/Series mainframe.
SOLA Development Studio	The SOLA Development Studio is a J2EE application that runs in a J2EE compliant server. You use a browser (IE 5.0 or higher) to access the SOLA Development Studio to create, publish, manage and monitor services.
SOLA Resource Manager	The SOLA Resource manager is a J2EE application that runs in a J2EE compliant server. You use a browser (IE 5.0 or higher) to access the SOLA Resource Manager to manage SOLA resources.
SOLA IMS enablement (optional for SOLA CICS Container)	The optional SOLA for IMS component includes the ability to expose transactions and subroutines that run in IMS/TM from the SOLA CICS Container. This component requires the installation of sample and subroutine driver programs and transactions in IMS/TM.

OVERVIEW OF SOLA INSTALLATION PROCESS

The SOLA installation process is designed to handle the following three scenarios:

1. Install SOLA 6.1 in a new customer installation
2. Install SOLA 6.1 as an upgrade to SOLA 5.1
3. Install SOLA 6.1 over a SOLA 6.0 installation.

Because the SOLA Directory has been redesigned, SOLA 6.1 ships with a Migration manual and a series of migration jobs to migrate the SOLA 5.1 Directory to the new SOLA 6.1 Directory. New customers can safely ignore the migration steps.

Upgrade customers (SOLA 5.1 to 6.1) will need to run the migration process referenced in the Migration Guide. The migration should be performed after the mainframe install is complete, and before you start on the SOLA Development Studio installation.

SOLA 6.0 testing sites will need to recycle the SOLA Server after they have run the migration batch jobs.

The first step in installing SOLA is to install the SOLA run-time container on the z/Series mainframe using SMP/E. Two run-time containers are provided: SOLA CICS Container (SOLA600) and SOLA IMS Container (SOLS600). Once the mainframe components are installed you complete the installation by customizing and running jobs from the SAMPLIB. Next you install the SOLA Development Studio into a J2EE application server, use the SOLA Development Studio to customize the SOLA installation and test the complete installation by running the sample applications included with the SOLA installation. Finally you install the SOLA Resource Manager to help you manage SOLA resources (Service Level Agreements, Policies, run-time containers, etc).

Note: The z/Series mainframe installation must be completed before starting the SOLA Development Studio installation.

IN THIS GUIDE

This guide includes the following chapters:

Chapter 1: "Installing the SOLA run-time on a z/Series mainframe" provides a list of steps for installing SOLA software components onto your z/Series mainframe.

Chapter 2: "Customizing the SOLA CICS Container on a z/Series mainframe" provides a list of steps for customizing SOLA CICS Container software components on your z/Series mainframe.

Chapter 3: "Customizing the SOLA IMS Container on a z/Series mainframe" provides a list of steps for customizing SOLA IMS Container software components on your z/Series mainframe.

Chapter 4: ""Installing the SOLA Development Studio" provides a list of steps for installing SOLA software components onto your J2EE server.

"Installing the SOLA Development Studio in WebSphere" provides a list of steps for installing SOLA software components into a WebSphere Application Server.

"Installing the SOLA Development Studio in WebLogic" provides a list of steps for installing SOLA software components into a Weblogic Application Server.

"Installing the SOLA Development Studio in Tomcat" provides a list of steps for installing SOLA software components into a Tomcat Application Server.

Chapter 5: "Customizing SOLA" provides a list of steps for installing and customizing the SOLA software components to make SOLA ready for use.

SYSTEM REQUIREMENTS

The following table lists the minimum system requirements for running SOLA 6.1.

Component Name	Requirement	
SOLA CICS Container (SOLA600)	<u>Hardware</u> z/Series class machine G4 processor or higher for encryption (if required)	
	<u>Operating System</u> <i>OS/390 & Z/OS</i> OS/390 Version 2 Release 9 or higher OR z/OS Version 1 Release 1 or higher	
	<i>CICS</i> CICS TS 1.3 or higher	
	<i>DB2</i> DB2 Version 7 or higher	
	<i>CICS CWS</i> CICS Web Support	
	<i>Sockets</i> CICS Sockets if outbound support and/or IMS is/are required	
	<i>3270</i> Bridge 3270 or Linkable Bridge for 3270 support	
	<i>MQ</i> MQ if MQ transport is required	
	<i>IMS</i> IMS version 9 or higher if SOA enablement of IMS transactions and/or subroutines is required	
	<i>UDDI / SOLA Dashboard</i> DB2 Version 8 or higher for UDDI inquiry	
	<i>Development Studio support</i> MVS FTP server	
	<i>Encryption</i> Integrated Cryptographic Service Facility (ICSF) for XML Encryption, XML Signature and outbound SSL (if required)	
	<i>Codepage Conversion</i> z/OS Conversion Services for conversion of Unicode	
SOLA IMS Container (SOLS600)	<u>Hardware</u> z/Series class machine G4 processor or higher for encryption (if required)	
	<u>Operating System</u> <i>OS/390 & Z/OS</i> OS/390 Version 2 Release 9 or higher OR z/OS Version 1 Release 1 or higher	

Component Name	Requirement	
	<i>DB2</i>	DB2 Version 7 or higher
	<i>IMS</i>	IMS version 9 or higher for SOA enablement of IMS transactions and/or subroutines
	<i>UDDI / SOLA Dashboard</i>	DB2 Version 8 or higher for UDDI inquiry
	<i>Development Studio support</i>	MVS FTP server
	<i>Encryption</i>	Integrated Cryptographic Service Facility (ICSF) for XML Encryption, XML Signature and outbound SSL (if required)
	<i>Codepage Conversion</i>	z/OS Conversion Services for conversion of Unicode
SOLA Development Studio	Any standard J2EE environment running a JRE (Java Runtime Environment) of 1.5 or above	
SOLA Resource Manager	Any standard J2EE environment running a JRE (Java Runtime Environment) of 1.5 or above	
Client	IE 5.5 and above	
Documentation	The SOLA product documentation is published in Portable Document Format (PDF) and requires Acrobat Reader 5.0 or above.	

CUSTOMER SUPPORT

SOA Software offers a variety of support services to our customers. The following options are available:

Support Options:	
Email (direct)	solasupport@soa.com .
Phone	1-877-337-8776 8:00 a.m. to 8:00 p.m. (PST)
Documentation Updates	Updates to SOLA product documentation are issued periodically, and are available by submitting an email request to solasupport@soa.com .

Chapter 1: Installing the SOLA run-time on a z/Series mainframe

INSTALLATION REQUIREMENTS AND CONSIDERATIONS

The following sections identify the system requirements for installing and activating SOLA for z/OS. The following terminology is used:

Driving system: the system used to install the program.

Target system: the system on which the program is installed.

The same system can be used as both a driving system and a target system.

DRIVING SYSTEM REQUIREMENTS

This section describes the environment of the driving system required to install SOLA for z/OS.

Machine Requirements

The driving system can run in any hardware environment that supports the required software.

Programming Requirements

Program Number	Product Name and Minimum VRM/Service Level
Any one of the following:	
5647-A01	OS/390 Version 2 Release 9 or higher
5694-A01	z/OS Version 1 Release 1 or higher
Any one of the following:	
5647-A01	OS/390 SMP/E Version 2 Release 9 or higher
5655-G44	SMP/E for z/OS and OS/390 Version 3 Release 1 or higher

Figure 1 Driving System Software Requirements

SOLA for z/OS uses REXX language ISPF edit macros during installation. These REXX language ISPF edit macros require OS/390 Version 2 Release 9 or higher or z/OS Version 1 Release 1 or higher.

TARGET SYSTEM REQUIREMENTS

This section describes the environment of the target system required to install and use SOLA for z/OS.

SOLA for z/OS installs in the MVS (Z038) SREL.

Machine Requirements

The target system can run in any hardware environment that supports the required software.

Programming Requirements

Mandatory Requisites

A mandatory requisite is defined as a product that is required without exception; this product either will not install or will not function unless this requisite is met. This includes products that are specified as REQs or PREs.

Program Number	Product Name and Minimum VRM/Service Level
Any one of the following:	
5647-A01	OS/390 Version 2 Release 9 or higher
5694-A01	z/OS Version 1 Release 1 or higher
All of the following:	
5675-DB2	IBM DB2 UDB Server for OS/390 Version 7.1 or higher
5696-234	High Level Assembler Version 1.2 or higher

Figure 2 Mandatory Requisites

Functional Requisites

A functional requisite is defined as a product that is not required for the successful installation of this product or for the basic function of the product, but is needed at run time for a specific function of this product to work.

Program Number	Product Name and Minimum VRM/Service Level	Function
5655-147	CICS TS Version 1 Release 3 or higher	SOLA CICS Container
5647-A01	ISPF Version 4.2.1 or higher	Installation
5647-A01	OS/390 Version 2.9 Cryptographic Services with fixes for APAR OW54083	WS-Security
5694-A01	z/OS Communications Server Version 1 Release 5	Development Studio and Sockets
5688-197 OR 5688-235	IBM COBOL for OS/390 and VM PL/I for MVS & VM	COMMAREA programs
5655-F10	WebSphere MQ for z/OS	MQ transport
5625-DB2	IBM DB2 UDB Server for OS/390 Version 8.1 or higher	UDDI Inquiry and SOLA Dashboard
5655-J38	IBM Information Management System (IMS), V9.1.0	SOLA IMS Container and/or IMS transactions and programs
	IMS Connect V9	IMS transactions and programs
	z/OS Conversion Services (part of base z/OS)	Unicode conversions

Figure 3 Functional Requisites

Toleration/Coexistence Requisites

A toleration/coexistence requisite is defined as a product which must be present on a sharing system. These systems can be other systems in a multi-system environment (not necessarily sysplex), a shared DASD environment (such as test and production), or systems that reuse the same DASD at different time intervals.

SOLA for z/OS has no toleration/coexistence requisites.

Incompatibility (Negative) Requisites

A negative requisite identifies products which must not be installed on the same system as this product.

There are no negative requisites for SOLA for z/OS.

Installing SOLA in any program product zones will not result in overlaying listed modules.

DASD Storage Requirements

SOLA for z/OS libraries can reside on all supported DASD types.

Figure 4 lists the total space required for each type of library.

Library	
Type	Total Space Required
Target	1,250 3390 tracks
Distribution	630 3390 tracks
Directory (DB2)	64 M Bytes (Estimated)
Statistics (DB2)	100 M Bytes (Estimated)

Figure 4 Total DASD space required by SOLA for z/OS

FMIDs Deleted

Installing SOLA for z/OS will result in the deletion of previous SOLA versions (FMIDs SOLA100 up to SOLA501).

Special Considerations

SOLA for z/OS has no special considerations for the target system.

INSTALLATION INSTRUCTIONS

This chapter describes the installation method and step-by-step procedures to install SOLA for z/OS. The next chapter describes further step-by-step procedures for customizing the functions of SOLA for z/OS.

SOLA for z/OS is distributed with the assumption that it will be installed in a separate SMP/E zone. It can be installed into an existing SMP/E zone with the restrictions described above.

When you have SMP/E installed SOLA for z/OS, refer to *Customizing the SOLA installation* on page 31 for testing and activating SOLA for z/OS.

The components of SOLA for z/OS

SOLA for z/OS comprises two products:

- SOLA CICS Container (SOLA600)
- SOLA IMS Container (SOLS600)

Each product has required features and optional features. These are explained here:

Required features:

SOLA CICS Container (SOLA600) Base

The base comprises all the basic functions for SOLA (the z/OS SOAP Stack). For SOLA CICS Container the base includes 3270 BMS support, COMMAREA support, Channels and Containers support, Callable support, Dynamic SQL support, DB2 Stored Procedure support and outbound support.

VSAM support has not been included in this release.

SOLA IMS Container (SOLS600) Base

The base comprises all the basic functions for SOLA (the z/OS SOAP Stack). For SOLA IMS Container the base includes IMS Transaction support, IMS Subroutine support and outbound support.

Optional Features

IMS support is an optional feature for SOLA CICS Container (SOLA600).

OVERVIEW OF SOLA FOR z/OS INSTALLATION

This section covers the installation of SOLA CICS Container (SOLA600) and SOLA IMS Container (SOLS600), which are referred to in this section jointly as SOLA for z/OS. Begin by choosing what product(s) you're installing on z/OS, because the installation process differs slightly between SOLA CICS Container and SOLA IMS Container. The installation process for SOLA for z/OS uses the SMP/E RECEIVE, APPLY, and ACCEPT commands to place SOLA for z/OS into the appropriate system libraries. If you need more information than is given in this document, refer to a full description of SMP/E in the IBM SMP/E User's Guide and SMP/E Reference books.

SMP/E is also used to apply preventive or corrective service after SOLA for z/OS has been installed.

This section covers the following topics. You are advised to read all these sections before you start the install of SOLA for z/OS.

Installation summary for SOLA for z/OS on page 8

Prepare to install SOLA for z/OS on page 9

Unzip the distribution files on page 11

Transmit the distribution files to the mainframe on page 13

Installation summary for SOLA for z/OS

The steps for installing SOLA for z/OS are summarized below. Sample JCL is provided for all these steps except 1 and 2.

Before installing SOLA for z/OS, you must choose names and values for a number of parameters. These names and values depend on the standards that apply to your z/OS environment. You also need to decide whether you want to install any of the optional features.

Review the sample installation worksheet provided on the distribution FTP site and customize it. The names and values you have chosen above will be used to modify the required JCL.

Set up an SMP/E environment. This environment must be tailored for SOLA for z/OS.

Add DDDEF statements to the SMP/E environment to define the target and distribution libraries to SMP/E (**only if installing into existing SMP/E zone**).

1. Allocate the target and distribution libraries.
2. Unzip the distribution files on the distribution FTP site.
3. Transmit the distribution files to the mainframe using FTP.
4. Run the SMP/E RECEIVE job to unload the required features from the distribution files.

5. Run the SMP/E APPLY job to apply the FMIDs.
6. Run the SMP/E ACCEPT job to accept the product.

When SOLA for z/OS has been installed, use SMP/E to install preventative or corrective service if required.

Prepare to install SOLA for z/OS

Before you install SOLA for z/OS, you must make the following decisions and preparations. The decisions that you make here affect the customization of the sample JCL provided with SOLA for z/OS. You should write down the appropriate values as you work through this section. Where applicable, space has been provided to note the values assigned.

Decide which parts of SOLA for z/OS you wish to install; the parts available are described in *The components of SOLA for z/OS* on page 7.

Ensure that you have the correct prerequisite products installed. For information about prerequisite products, and the levels of these products required to install and use SOLA for z/OS, see *System Requirements* on page xii.

Plan the environment that you are going to install SOLA for z/OS into. You need to decide:

Space requirements (see “*DASD Storage Requirements*” on page 6)

High-level qualifiers to use (see “*High-level qualifiers*” on page 9)

SMP/E environment: creating a new or using an existing SMP/E environment (see “*SMP/E environment*” on page 9).

High-level qualifiers

You must customize the installation JCL to specify the high-level qualifiers used by SOLA for z/OS. Use the attached worksheet (WKSHEET.doc) to assist with this task and then make the changes in the SOLAEDT edit macro. Any high-level qualifiers that do not already exist must be defined to RACF or any other security product that you are using, and have ALIAS definitions in the master catalog.

SMP/E environment

You must decide whether to install SOLA for z/OS into a new or an existing SMP/E environment. Remember that installing a new release of SOLA for z/OS into the same SMP/E zones as an existing release will cause the earlier level to be deleted from both the SMP/E zones and the existing release data sets.

If you are going to install into the same SMP/E zones, preserve your current release while installing and testing the new release. You can achieve this by copying the existing release data sets into data sets with different names. Do not rename or delete the current data sets because they need to be available to SMP/E for delete processing during the installation of the new release. If they are not available, the APPLY and ACCEPT steps will fail with a return code of 12. After running a successful ACCEPT of the SOLA for z/OS products, the previous release data sets are no longer required and can be deleted when testing of the new release has been completed.

If you use existing SMP/E data sets, you need to know the names of your target zone and distribution zone, and the data set name of your global CSI.

Note: If you are installing into existing zones:

The PEMAX options entry must be at least 4500 or left to default.

The DSSPACE options entry must specify at least 100 directory blocks.

You must decide upon the names for the SMP/E zones (see WKSHEET.doc).

UNZIP THE DISTRIBUTION FILES

The root directory of the distribution package contains five directories:

	Documentation
	IDE
	PTF
	Utilities
	zSeries
	ReadMe.txt
	WRKSHEET IDE.doc
	WRKSHEET.doc

The zSeries directory contains two sub directories, one for SOLA CICS Container and the other for SOLA IMS Container. Each sub directory contains smpe.zip, which contains all the distribution files you need for SOLA's SMP/E install.

For SOLA CICS Container, unzipping the smpe.zip file will create the following files:

```
$putfiles.cmd      <== windows bat file to start ftp transfer  
$putfiles.scp     <== contains ftp commands to upload files  
  
SOLA600.F1.XMI  
  
SOLA600.F2.XMI  
  
SOLA600.F3.XMI  
  
SOLA600.F4.XMI  
  
SOLA600.F5.XMI  
  
SOLA600.SMPPTFIN.XMI  
  
SOLA600.INSTLIB.XMI  <== installation jobs  
SOLA600.JCL.XMI      <== first two installation jobs+WORKSHEET  
SOLA600.TXT          <== first installation job
```

Note: .XMI files are TSO XMITted PDS data sets.

For SOLA IMS Container, unzipping the smpe.zip file will create the following files:

```
$putfiles.cmd      <== windows bat file to start ftp transfer  
$putfiles.scp     <== contains ftp commands to upload files
```

SOLS600.F1.XMI
SOLS600.F2.XMI
SOLS600.F3.XMI
SOLS600.F4.XMI
SOLS600.F5.XMI
SOLS600.SMPPTFIN.XMI
SOLS600.INSTLIB.XMI <== installation jobs
SOLS600.JCL.XMI <== first two installation jobs+WORKSHEET
SOLS600.TXT <== first installation job

Note: .XMI files are TSO XMITted PDS data sets.

TRANSMIT THE DISTRIBUTION FILES TO THE MAINFRAME

Transmit the distribution files to the mainframe (SOLA600)

Edit \$putfiles.cmd using Windows Notepad file editor:

The file contains a single record, as follows:

```
ftp -v -n -i <MainFrameFTPServerName> <$putfiles.scp
```

Change *<MainFrameFTPServerName>* to your mainframe ftp server FQDN. For example, if your mainframe FTP Server FQDN is MF.TESTSVR.COM then change *<MainFrameFTPServerName>* to MF.TESTSVR.COM. Make sure you remove the < and > characters. After your changes the record would be:

```
ftp -v -n -i MF.TESTSVR.COM <$putfiles.scp
```

Edit \$putfiles.scp using Windows Notepad file editor:

The file contents are shown below:

```
user <userName> <passWord>
```

```
type image
```

```
QUOTE SITE REC=FB LR=80 BLK=3120
```

```
put 'SOLA600.f1.xmi'      '<xmihlq>.SOLA600.f1.xmi'
```

```
put 'SOLA600.f2.xmi'      '<xmihlq>.SOLA600.f2.xmi'
```

```
put 'SOLA600.f3.xmi'      '<xmihlq>.SOLA600.f3.xmi'
```

```
put 'SOLA600.f4.xmi'      '<xmihlq>.SOLA600.f4.xmi'
```

```
put 'SOLA600.f5.xmi'      '<xmihlq>.SOLA600.f5.xmi'
```

```
put 'SOLA600.INSTLIB.xmi'  '<xmihlq>.SOLA600.INSTLIB.xmi'
```

```
put 'SOLA600.JCL.xmi'     '<xmihlq>.SOLA600.JCL.xmi'
```

```
put 'SOLA600.SMPPTFIN.xmi' '<xmihlq>.SOLA600.SMPPTFIN.xmi'
```

```
ASCII
```

```
QUOTE SITE REC=FB LR=80 BLK=27920
```

```
put 'SOLA600.txt'         '<xmihlq>.SOLA600.txt'
```

Change *<userName>* & *<passWord>* to a valid RACF userid and password. These fields are not case sensitive. Make sure you remove the < and > characters.

Change all references to *<xmihlq>* to the transmit(xmit) high-level-qualifier that you chose (see wrksheet.doc). Make sure you remove the < and > characters.

Transfer all files from distribution package to the mainframe by executing \$putfiles.cmd or manually transmitting them.

Note: it may be beneficial to pre-allocate the datasets <xmihlq>.SOLA600. on the mainframe before starting the FTP.*

If you choose to manually transmit the files, you will need to:

Allocate all .XMI files with the DCB attributes of LRECL=80, BLKSIZE=3120 and transmit as binary

Allocate <xmihlq>.SOLA600.TXT with the DCB attributes of LRECL=80 and transmit as ASCII.

We recommend that you create an alias for the high level qualifier (hlq) for SOLA V6R0M0 installation files. This will catalog all these files in the USER ICF catalog. Customize this sample job to define such an alias:

```
//STEP010 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//SYSIN     DD *
DEFINE ALIAS (NAME(disthlq) -
RELATE(your.user.catalog))-
CATALOG(your.master.catalog/password)
/*
//
```

Note that "disthlq" should correspond to parameter <disthlq>, which will be set in the installation worksheet later.

If you choose not to create a catalog alias then select a value for <disthlq> that you have write access to.

Transmit the distribution files to the mainframe (SOLS600)

Edit \$putfiles.cmd using Windows Notepad file editor:

The file contains a single record, as follows:

```
ftp -v -n -i <MainFrameFTPServerName> < $putfiles.scp
```

Change *<MainFrameFTPServerName>* to your mainframe ftp server FQDN. For example, if your mainframe FTP Server FQDN is MF.TESTSVR.COM then change *<MainFrameFTPServerName>* to MF.TESTSVR.COM. Make sure you remove the < and > characters. After your changes the record would be:

```
ftp -v -n -i MF.TESTSVR.COM < $putfiles.scp
```

Edit \$putfiles.scp using Windows Notepad file editor:

The file contents are shown below:

```
user <userName> <passWord>
```

```
type image
```

```
QUOTE SITE REC=FB LR=80 BLK=3120
```

```
put 'SOLS600.f1.xmi'      '<xmihlq>.SOLS600.f1.xmi'
```

```
put 'SOLS600.f2.xmi'      '<xmihlq>.SOLS600.f2.xmi'
```

```
put 'SOLS600.f3.xmi'      '<xmihlq>.SOLS600.f3.xmi'
```

```
put 'SOLS600.f4.xmi'      '<xmihlq>.SOLS600.f4.xmi'
```

```
put 'SOLS600.f5.xmi'      '<xmihlq>.SOLS600.f5.xmi'
```

```
put 'SOLS600.INSTLIB.xmi'  '<xmihlq>.SOLS600.INSTLIB.xmi'
```

```
put 'SOLS600.JCL.xmi'     '<xmihlq>.SOLS600.JCL.xmi'
```

```
put 'SOLS600.SMPPTFIN.xmi' '<xmihlq>.SOLS600.SMPPTFIN.xmi'
```

```
ASCII
```

```
QUOTE SITE REC=FB LR=80 BLK=27920
```

```
put 'SOLS600.txt'         '<xmihlq>.SOLS600.txt'
```

Change *<userName>* & *<passWord>* to a valid RACF userid and password. These fields are not case sensitive. Make sure you remove the < and > characters.

Change all references to *<xmihlq>* to the transmit(xmit) high-level-qualifier that you chose (see wrksheet.doc). Make sure you remove the < and > characters.

Transfer all files from distribution package to the mainframe by executing \$putfiles.cmd or manually transmitting them.

Note: it may be beneficial to pre-allocate the datasets <xmihlq>.SOLA600. on the mainframe before starting the FTP.*

If you choose to manually transmit the files, you will need to:

Allocate all .XMI files with the DCB attributes of LRECL=80, BLKSIZE=3120 and transmit as binary

Allocate <xmihlq>.SOLS600.TXT with the DCB attributes of LRECL=80 and transmit as ASCII.

We recommend that you create an alias for the high level qualifier (hlq) for SOLA SOLS600 installation files. This will catalog all these files in the USER ICF catalog. Customize this sample job to define such an alias:

```
//STEP010 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//SYSIN     DD *
DEFINE ALIAS(NAME(disthlq) -
RELATE(your.user.catalog))-
CATALOG(your.master.catalog/password)
/*
//
```

Note that "disthlq" should correspond to parameter <disthlq>, which will be set in the installation worksheet later.

If you choose not to create a catalog alias then select a value for <disthlq> that you have write access to.

CREATE THE INSTALL LIBRARY

Create the install library (SOLA600)

Customize SOLA600.TXT. This job will TSO RECEIVE <xmihlq>.SOLA600.JCL.XMI and create the installation library <disthlq>.SOLA600.JCL.

<disthlq>.SOLA600.JCL will contain these members:

DEFALIAS - optional job to define ALIAS for <disthlq> (used only if distribution files are to be cataloged outside of the MASTER CATALOG)

TSORECVE - TSO RECEIVE job - a job to unpack the XMITted distribution files.

WRKSHEET - a list of parameters to be customized to your installation standards.

Create the install library (SOLS600)

Customize SOLS600.TXT. This job will TSO RECEIVE <xmihlq>.SOLS600.JCL.XMI and create the installation library <disthlq>.SOLS600.JCL.

<disthlq>.SOLS600.JCL will contain these members:

DEFALIAS - optional job to define ALIAS for <disthlq> (used only if distribution files are to be cataloged outside of the MASTER CATALOG)

TSORECVE - TSO RECEIVE job - a job to unpack the XMITted distribution files.

WRKSHEET - a list of parameters to be customized to your installation standards.

CUSTOMIZE WRKSHEET FOR YOUR ENVIRONMENT

Customize WRKSHEET for your environment (SOLA600)

Review and customize WRKSHEET in <disthlq>.SOLA600.JCL to comply with your installation standards.

Parameter Name in the sample WRKSHEET provided	Description	Notes/Examples	Your Values
<jobcrd0>	Job card information (1st line)		
<jobcrd1>	Job card information (2nd line)		
<smpehlp>	SMP/E high level qualifier	"SOLA.GLOBAL" will result in "SOLA.GLOBAL.CSI"	
<smptzhlp>	SMP/E target zone high level qualifier	"SOLA.SOLA600.TZN" will result in "SOLA.SOLA600.TZN.CSI"	
<smpdzhlq>	SMP/E distribution zone high level qualifier	"SOLA.SOLA600.DZN" will result in "SOLA.SOLA600.DZN.CSI"	
<xmihlp>	SMP/E xmit high level qualifier	"SOLAXMI" will result in "SOLAXMI.SOLA600.F1.XMI" etc.	
<disthlq>	Temporary distribution high level qualifier (used by TSO RECEIVE)	Must be a single hlq. Cannot be a compound hlq (for example SYSE.SOLA is not allowed by SMP/E). "SOLA" will result in "SOLA.SOLA600.F1"	
<tlibhlq>	Target library high level qualifier	"SOLA.V6R0M1" will result in "SOLA.V6R0M1.LOADLIB"	
<dlibhlq>	Distribution library high level qualifier	SOLA.V6R0M1 will result in "SOLA.V6R0M1.AMODLIB"	
<tlibvol>	Target library VOLSER	SOAP00	
<dlibvol>	Distribution library VOLSER	SOAP00	
<smpevol>	SMP/E library VOLSER	SOAP00	
<smptlbpr>	SMP/E Rel Files prefix	SOLA.SMPTLIB will result in "SOLA.SMPTLIB.SOLA600.F1"	
<dclas>	Data class (optional)	Optional	
<mclas>	Management class (optional)	Optional	
<sclas>	Storage class (optional)	Optional	
<tgtblksize>	FB datasets blksize	27920	

Parameter Name in the sample WRKSHEET provided	Description	Notes/Examples	Your Values
<diskunit>	Disk unit name	SYSDA	
<smpetlib>	SMP/E Target Zone name	SOLATGT	
<smpedlib>	SMP/E Distribution Zone name	SOLADIS	
<sceelked.dsnm>	LE library	SYS1.SCEELKED	
<sezatcp.dsnm>	TCP/IP library	SYS1.SEZATCP	
<scsfmod0.dsnm>	ICSF library	SYS1.SCSFMOD0	
<csslib..dsnm>	CSS Library	SYS1.CSSLIB	
<db2SDSNLOAD>	DB2 SDSNLOAD Library	SYSAPF.DBMS.DB2GR0Q.SDSNL OAD	
<db2RunLoad>	DB2 Run load module library	DB2GR0Q.RUNLIB.LOAD	
<subSystem>	DB2 Subsystem	GR0Q	
<collection>	DB2 Package collection	XML	
<plan>	DB2 PLAN name	XMLPLAN	
<qualifier>	DB2 Qualifier	SOLA600	
<Oldqualifier>	SOLA 5.1 Qualifier (Migration Only)	SOLAQUAL	
<indexBP>	DB2 Index Buffer Pool	BP3	
<tableBP>	DB2 Tablespace BufferPool	BP2	
<BP32K2>	DB2 32K BufferPool	BP32K2	
<stoGroup>	DB2 Stogroup	SGXMLSMS	
<vCat>	DB2 VCAT (used to create stogroup)	SOLA	
<group>	RDO Group name	SOLAGR	
<list>	List name	SOLA	
<cicsLOAD>	CICS LOAD library (SOLA CICS Container only)	CICS.TEST.WQ62.SDFHLOAD	
<csdFile>	CICS CSD File (SOLA CICS Container only)	TVWQCICS.CICSTS22.DFHCS	
<vsamUMT>	UMT VSAM File name (used by SOLA CICS Container only)	SOLA.VSAM.MONTR.QCICST	
<vsamCMT>	CMT VSAM File (used by SOLA CICS Container for Identity mapping)	SOLA.VSAM.MAPPING.QCICST	
<vol>	Volume where VSAM file will be allocated	SOAP00	

Customize WRKSHEET for your environment (SOLS600)

Review and customize WRKSHEET in <*disthlq*>.SOLS600.JCL to comply with your installation standards.

Parameter Name in the sample WRKSHEET provided	Description	Notes/Examples	Your Values
<jobcrd0>	Job card information (1st line)		
<jobcrd1>	Job card information (2nd line)		
<smpehlq>	SMP/E high level qualifier	"SOLA.GLOBAL" will result in "SOLA.GLOBAL.CSI"	
<smptzhlpq>	SMP/E target zone high level qualifier	"SOLA.SOLS600.TZN" will result in "SOLA.SOLS600.TZN.CSI"	
<smpdzhlq>	SMP/E distribution zone high level qualifier	"SOLA.SOLS600.DZN" will result in "SOLA.SOLS600.DZN.CSI"	
<xmihlq>	SMP/E xmit high level qualifier	"SOLAXMI" will result in "SOLAXMI.SOLS600.F1.XMI" etc.	
<disthlq>	Temporary distribution high level qualifier (used by TSO RECEIVE)	Must be a single hlq. Cannot be a compound hlq (for example SYSE.SOLA is not allowed by SMP/E). "SOLA" will result in "SOLA.SOLS600.F1"	
<tlibhlq>	Target library high level qualifier	"SOLA.SOLS600" will result in "SOLA.SOLS600.LOADLIB"	
<dlibhlq>	Distribution library high level qualifier	SOLA.SOLS600 will result in "SOLA.SOLS600.AMODLIB"	
<tlibvol>	Target library VOLSER	SOAP00	
<dlibvol>	Distribution library VOLSER	SOAP00	
<smpevol>	SMP/E library VOLSER	SOAP00	
<smptlbpr>	SMP/E Rel Files prefix	SOLA.SMPTLIB will result in "SOLA.SMPTLIB.SOLS600.F1"	
<dclas>	Data class (optional)	Optional	
<mclas>	Management class (optional)	Optional	
<sclas>	Storage class (optional)	Optional	
<tgtdblk>	FB datasets blksize	27920	
<diskunit>	Disk unit name	SYSDA	
<smpetlib>	SMP/E Target Zone name	SOLATGT	
<smpedlib>	SMP/E Distribution Zone name	SOLADIS	

Parameter Name in the sample WRKSHEET provided	Description	Notes/Examples	Your Values
<sceelked.dsnm>	LE library	SYS1.SCEELKED	
<sezatcp.dsnm>	TCP/IP library	SYS1.SEZATCP	
<scsfmod0.dsnm>	ICSF library	SYS1.SCSFMOD0	
<csslib..dsnm>	CSS Library	SYS1.CSSLIB	
<db2SDSNLOAD>	DB2 SDSNLOAD Library	SYSAPF.DBMS.DB2GR0Q.SDSNL OAD	
<db2RunLoad>	DB2 Run load module library	DB2GR0Q.RUNLIB.LOAD	
<subSystem>	DB2 Subsystem	GR0Q	
<collection>	DB2 Package collection	XML	
<plan>	DB2 PLAN name	XMLPLAN	
<qualifier>	DB2 Qualifier	SOLA600	
<Oldqualifier>	SOLA 5.1 Qualifier (Migration Only)	SOLAQUAL	
<indexBP>	DB2 Index Buffer Pool	BP3	
<tableBP>	DB2 Tablespace BufferPool	BP2	
<BP32K2>	DB2 32K BufferPool	BP32K2	
<stoGroup>	DB2 Stogroup	SGXMLSMS	
<vCat>	DB2 VCAT (used to create stogroup)	SOLA	
<TCPAddressSpace>	TCPIP jobname	TCPIP	
<TCPPort>	TCPIP Port Number (SOLA Started Task Listener Port). 5 digit number prefixed with Zeroes	03800	
<SOLAsysid>	4 Character SYSID for the SOLA Started task	SOL1 Must be unique within a Sysplex	
<Qmgr>	4 Character MQ Subsystem ID	CSQ7	
<MQRequestQ>	Max 48 Character SOLA Request Queue	SOLA.REQUEST.QUEUE	

RECEIVE THE DISTRIBUTION FILES

Receive the distribution files (SOLA600)

Customize the JOBCARD, *<disthlq>* and *<xmihlq>* parms in the *<xmihlq>.SOLA600.txt* dataset and submit the job.

This will receive the following datasets:

<disthlq>.SOLA600.JCL

Inside this dataset will be a member named TSORECVE. This job will TSO RECEIVE the distribution files. Customize this JCL to receive the remaining distribution files.

The following datasets will be created:

<disthlq>.SOLA600.F1

<disthlq>.SOLA600.F2

<disthlq>.SOLA600.F3

<disthlq>.SOLA600.F4

<disthlq>.SOLA600.F5

<disthlq>.SOLA600.SMPPTFIN

<disthlq>.SOLA600.INSTLIB

Receive the distribution files (SOLS600)

Customize the JOBCARD, *<disthlq>* and *<xmihlq>* parms in the *<xmihlq>.SOLS600.txt* dataset and submit the job.

This will receive the following datasets:

<disthlq>.SOLS600.JCL

Inside this dataset will be a member named TSORECVE. This job will TSO RECEIVE the distribution files. Customize this JCL to receive the remaining distribution files.

The following datasets will be created:

<disthlq>.SOLS600.F1

<disthlq>.SOLS600.F2

<disthlq>.SOLS600.F3

<disthlq>.SOLS600.F4

<disthlq>.SOLS600.F5

<disthlq>.SOLS600.SMPPTFIN

<disthlq>.SOLS600.INSTLIB

CUSTOMIZE THE SAMPLE INSTALLATION JCL

Customize the sample installation JCL (SOLA600)

Editing the SOLAEDT macro

Customize SOLAEDT in *<disthlq>.SOLA600.INSTLIB* with values from the WRKSHEET created in Customize WRKSHEET for your environment (SOLA600)on page 18. SOLAEDT is a Rexx Exec. Ensure that single quotes (' ') and double quotes (" ") are not altered.

Adding the edit macro SOLAEDT to the sysproc concatenation

SOLAEDT is a Rexx Exec. Rexx Execs can execute from either SYSPROC or SYSEXEC. Copy SOLAEDT to a dataset in your SYSPROC or SYSEXEC concatenation.

To find out what datasets are allocated to SYSPROC or SYSEXEC for your TSO session you can issue the TSO ISRDDN command from the ISPF command line and then find SYSPROC or SYSEXEC in the DDname column. Choose a dataset that you are authorized to write to and copy SOLAEDT from *<disthlq>.SOLA600.INSTLIB* into that dataset.

Executing the SOLAEDT macro

Customize *<disthlq>.SOLA600.SMPPTFIN(SOLA600)* by executing the SOLAEDT edit macro.

SOLAEDT is executed by typing SOLAEDT on the command line while you are editing a dataset with ISPF edit. For example, to customize the dataset *<disthlq>.SOLA600.SMPPTFIN(SOLA600)* you would type SOLAEDT on the command line while editing *<disthlq>.SOLA600.SMPPTFIN(SOLA600)* using ISPF edit.

The following jobs in *<disthlq>.SOLA600.INSTLIB* can be customized by executing the SOLAEDT edit macro while in ISPF edit.

SMPECSI

SMPALLOC

RECVFUNC

APPLFUNC

ACCPFUNC

SOLAEDT will fill in any user defined values as defined in WRKSHEET specified in Transmit the distribution files to the mainframe (SOLA600)on page 13.

Customize the sample installation JCL (SOLS600)

Editing the SOLAEDT macro

Customize SOLAEDT in *<disthlq>.SOLS600.INSTLIB* with values from the WRKSHEET created in Customize WRKSHEET for your environment (SOLS600) on page 20. SOLAEDT is a Rexx Exec. Ensure that single quotes (' ') and double quotes (" ") are not altered.

Adding the edit macro SOLAEDT to the sysproc concatenation

SOLAEDT is a Rexx Exec. Rexx Execs can execute from either SYSPROC or SYSEXEC. Copy SOLAEDT to a dataset in your SYSPROC or SYSEXEC concatenation.

To find out what datasets are allocated to SYSPROC or SYSEXEC for your TSO session you can issue the TSO ISRDDN command from the ISPF command line and then find SYSPROC or SYSEXEC in the DDname column. Choose a dataset that you are authorized to write to and copy SOLAEDT from *<disthlq>.SOLS600.INSTLIB* into that dataset.

Executing the SOLAEDT macro

Customize *<disthlq>.SOLS600.SMPPTFIN(SOLS600)* by executing the SOLAEDT edit macro.

SOLAEDT is executed by typing SOLAEDT on the command line while you are editing a dataset with ISPF edit. For example, to customize the dataset *<disthlq>.SOLS600.SMPPTFIN(SOLS600)* you would type SOLAEDT on the command line while editing *<disthlq>.SOLS600.SMPPTFIN(SOLS600)* using ISPF edit.

The following jobs in *<disthlq>.SOLS600.INSTLIB* can be customized by executing the SOLAEDT edit macro while in ISPF edit.

SMPECSI

SMPECSI1

SMPALLOC

SMPALLC1

RECVFUNC

APPLFUNC

ACCPFUNC

SOLAEDT will fill in any user defined values as defined in WRKSHEET specified in Customize WRKSHEET for your environment (SOLS600) on page 20.

USE SMP/E TO INSTALL SOLA FOR Z/OS

Use SMP/E to Install SOLA CICS Container (SOLA600)

Installing into your SMP/E Environment

Run all jobs in *<disthlq>.SOLA600.INSTLIB* in the following order:

Job	Description
SMPECSI	to create the SMP/E database for SOLA software
SMPALLOC	to create target and distribution SMP/E files
RECVFUNC	to SMP/E RECEIVE function SOLA600 (SOLA V6R0M0) into your GLOBAL zone. Expect RC=0
APPLFUNC	to SMP/E APPLY CHECK SOLA function. Remove CHECK keyword if satisfied with APPLY CHECK processing in order to update your libraries. Expect RC=0
ACCPFUNC	to SMP/E ACCEPT CHECK SOLA function. Remove CHECK keyword if satisfied with ACCEPT CHECK processing in order to update your libraries. Expect RC=0

Installing Additional SMP/E Maintenance

SOLA is distributed with preventive or corrective maintenance which has been developed after base function packaging. This maintenance is distributed in *<disthlq>.SOLA600.SMPPTFIN*. We strongly recommend implementing this additional maintenance. Your distribution package will contain the following two JCLs: "Receive ALL PTFs.txt" and "Apply All PTFs.txt". Customize these two JCLs using SOLAEDT Macro and submit them to receive and apply all of the PTFs.

SOLA600 PTFs follow this naming convention:

SFXxxxx

Where y is the SOLA version number and xxx is the PTF number, for example, SFX6001 is SOLA CICS Container v6 PTF number 1.

You may reference the *<disthlq>.SOLA600.SMPPTFIN* data set and read the technical description of each PTF to decide if the particular PTF is applicable to your environment.

Receiving Additional SMP/E Maintenance

RECVPTF in *<disthlq>.SOLA600.INSTLIB* is provided to SMP/E receive any additional/future preventative maintenance distributed with the base function. Customize this job by executing the SOLAEDT macro while in ISPF edit mode.

Applying Additional SMP/E Maintenance

APPLYPTF in *<disthlq>.SOLA600.INSTLIB* is provided to apply any additional/future preventative or corrective maintenance.

This job will SMP/E APPLY CHECK any maintenance to the SOLA function. Remove the CHECK keyword if you're satisfied with APPLY CHECK processing in order to update your libraries.

Use SMP/E to Install SOLA IMS Container (SOLS600)

Installing into your SMP/E Environment

Run all jobs in <disthlq>.SOLS600.INSTLIB in the following order:

Job	Description
SMPECSI (or) SMPECSI1	Create the SMP/E database for SOLA software in a dedicated SMP/e Env If your site has already installed SOLA CICS Container and you want to install SOLA IMS Container in the same Global zone then use this JCL to alter the SMPE Database for SOLA IMS Container
SMPALLOC (or) SMPALLC1	Create target and distribution SMP/E files in a dedicated SMP/e Env If your site has already installed SOLA CICS Container and you want to install SOLA IMS Container in the same Global zone then use this JCL to allocate the new SOLA IMS Container target/distribution libraries
RECVFUNC	SMP/E RECEIVE function SOLS600 (SOLA SOLS600) into your GLOBAL zone. Expect RC=0
APPLFUNC	MP/E APPLY CHECK SOLA function. Remove CHECK keyword if satisfied with APPLY CHECK processing in order to update your libraries. Expect RC=4
ACCPFUNC	SMP/E ACCEPT CHECK SOLA function. Remove CHECK keyword if satisfied with ACCEPT CHECK processing in order to update your libraries. Expect RC=0

Installing Additional SMP/E Maintenance

SOLA is distributed with preventive or corrective maintenance which has been developed after base function packaging. This maintenance is distributed in *<disthlq>.SOLS600.SMPPTFIN*. We strongly recommend implementing this additional maintenance. Your distribution package will contain the following two JCLs: "Receive ALL PTFs.txt" and "Apply All PTFs.txt". Customize these two JCLs using SOLAEDT Macro and submit them to receive and apply all of the PTFs.

SOLS600 PTFs follow this naming convention:

SFSyxxx

Where y is the SOLA version number and xxx is the PTF number, for example, SFS6001 is SOLA IMS Container v6 PTF number 1.

You may reference the *<disthlq>.SOLS600.SMPPTFIN* data set and read the technical description of each PTF to decide if the particular PTF is applicable to your environment.

Receiving Additional SMP/E Maintenance

RECVPTF in *<disthlq>.SOLS600.INSTLIB* is provided to SMP/E receive any additional/future preventative maintenance distributed with the base function. Customize this job by executing the SOLAEDT macro while in ISPF edit mode.

Applying Additional SMP/E Maintenance

APPLYPTF in *<disthlq>.SOLS600.INSTLIB* is provided to apply any additional/future preventative or corrective maintenance.

This job will SMP/E APPLY CHECK any maintenance to the SOLA function. Remove the CHECK keyword if you're satisfied with APPLY CHECK processing in order to update your libraries.

Chapter 2: Customizing SOLA CICS Container on a z/Series mainframe (SOLA600)

CUSTOMIZING THE SOLA INSTALLATION

The following sections describe the steps necessary to customize the sample members and jobs in the `<tlibhlq>.SAMPLIB` and then run those jobs to:

- Customize the SAMPLIB
- Run jobs in SAMPLIB to:
 - Create the SOLA DB2 database
 - Bind the plan and all of the packages in the SOLA collection
 - Create the VSAM file used by the SOLA logger
 - Define the CICS table entries in the CICS CSD.
 - Optional: Define the CICS Analyzer to use with SOLA. This is only needed if you have a special security requirement (more details later in this section).
 - Optional: If you are upgrading from SOLA 5.1 then you will need to refer to the SOLA 6.1 Migration Guide for help in migrating the SOLA 5.1 Directory to the new SOLA 6.1 Directory.

CUSTOMIZING THE `<TLIBHLQ>.SAMPLIB`

Refer to the #README member of `<tlibhlq>.SAMPLIB`.

Run the SOLAEDT macro on the following members of `<tlibhlq>.SAMPLIB`:

Member	Description
BIND, BIND1	Plan and Package bind commands
BINDMIG	Package bind commands for SOLA5.1->SOLA6.1 Migration Programs. Not required for new installations.

Member	Description
BINDJCL	JCL to bind the plan and packages
BINDJCLM	JCL to bind SOLA5.1->SOLA6.1 Migration packages. Not required for new installations.
CSD, CSD1, CSD2	SOLA CICS Container only. PPT, PCT, FCT, TDQ and RCT/DB2 Entry definitions for SOLA executables
CSDJCL	SOLA CICS Container only. JCL to define the CSD entries
DCL	DB2 Grant statements
DDL	DDL to create the DB2 STOGROUP (optional), database, tablespaces, tables, indexes and aliases.
DDLMIG	DDL to create SOLA5.1 table aliases that is used for SOLA5.1->SOLA6.1 migration. Not required for new installations.
DDLJCL	JCL to create the DB2 entries
DDLJCLM	JCL to create SOLA5.1 table alias entries for SOLA5.1->SOLA6.1 migration. Not required for new installations.
IDCAMJCL	SOLA CICS Container only. JCL to define the VSAM files used by the SOLA logger & identity mapping functionality.
IMAPLOAD	Job to populate table TBXMLMFD for identity mapping.
IMSGEN	Macro instructions to define the IMS subroutine driver and sample transactions.
PSBGEN	Macro instructions to define the PSBs for the subroutine driver programs and the sample program.

CREATING THE DATABASE WITH DDLJCL

You create the SOLA directory by running the job DDLJCL. This job requires DB2 database privileges and may need to be run by a DBA. In order to successfully run the job you must have appropriate DB2 authority to create a STOGROUP (optional), a DATABASE (DBXML002), TABLESPACE, TABLE, INDEX and ALIAS.

The SOLA directory is defined in a single database DBXML002. You can create this database in its own STOGROUP or you can use an existing STOGROUP. If you want to use a STOGROUP specifically for DBXML002 then uncomment the "CREATE STOGROUP" statement at the beginning of member DDL.

The SOLA Database DBXML002 consists of at least 28 tables. Each table is created in its own unique tablespace. There are at least 47 indexes. At least 28 ALIASes are defined.

Job DDLJCL should execute with a return code of zero.

BINDING THE PLAN AND PACKAGES WITH BINDJCL

You bind the SOLA packages and plan by running jobs BINDJCL.

The person running these jobs must have appropriate authority to select, update, insert and delete the tables in the DBXML002 database.

There are at least 20 package bind cards in member BIND. All of these packages are bound into a single collection <collection>. There is a single plan bind statement for plan <plan> in member BIND1 which includes all packages in PKLIST <collection>.

Job BINDJCL should execute with a return code of zero.

CREATING THE <VSAMUMT> AND <VSAMCMT> FILES WITH IDCAMJCL

SOLA CICS Container uses CICS Data Tables for caching; a User Maintained Data Table (UMT) and a CICS Maintained Data Table (CMT). You define the UMT and CMT by running job IDCAMJCL.

The UMT is used to buffer all of the execution statistics collected by the SOLA run-time engine. These statistics are flushed from the <vsamUMT> file by a background task that periodically reads from the file and copies the data to a DB2 table <qualifier>.TBXMLMON.

The CMT is used to store the User Identity mapping data from the database to be used by the SOLA Analyzer (runtime component). A background task periodically checks updates to the mapping data in the database and reflects the changes to the CMT.

Job IDCAMJCL should execute with a return code of zero.

Note: Each TOR/WOR region with a SOLA runtime should have a dedicated XMLKSMAP CMT VSAM file. Sharing of the XMLKSMAP VSAM file is not supported.

POPULATE THE IDENTITY MAPPING TABLE USING JOB IMAPLOAD

Use job IMAPLOAD to populate table TBXMLMFD. The following are the details of the data stored in TBXMLMFD:

- TOR_SYS_ID: char(4), SYSID of WOR region.
- ID_MAP_TYP: char(1), type of mapping , value 'O' is for identity mapping.
- SUBJECT_ID: varchar(255),subject ID.
- RACF_USR_ID: char(8), RACF mainframe user id.
- TRN_ID: transid , optional, to override XML transaction.
- TMPLT_ID: template Id, optional, to override identity mapping template XML#DAN, for future use.
- DEPLOY_TS: when this entry was created in the table.

DEFINE THE CICS CSD ENTRIES WITH CSDJCL

You create the SOLA CICS CSD entries by running job CSDJCL.

Because SOLA CICS Container executes in a CICS region it requires CICS table entries (PCT, PPT, TDQ, RCT and FCT). The CICS CSD entries are shipped with default entries and should be customized based on your setup. All PCT definitions are delivered in samplib member CSD1 with default TRANCLASS(DFHTCL00). This needs to be customized based on your installation standards. The job CSDJCL creates these entries.

Jobs CSDJCL should execute with a return code of zero.

DEFINE THE CICS ANALYZER TO USE WITH SOLA (OPTIONAL)

By default SOLA CICS Container runs with the default CICS Analyzer DFHWBADX. With this analyzer, which runs under transaction CWXN, it isn't possible to run your user transactions under a SAF ID (such as RACF, ACF2 or Top-Secret controlled Ids).

For security reasons the SOLA Development Studio uses the WS-Security specification to pass user credentials between the Development Studio and the SOLA CICS Container back-end. In order to run the back-end transactions under the SAF ID that's passed up in the SOAP message then you will need to use the SOLA XMLPCAN Analyzer, as detailed below:

XMLPCAN: Will run the user transaction under the SAF ID that is passed in the WS-Security Header.

In order to use this special purpose analyzer, you need to alter your TCPIPS definition on CICS as follows (This can be done by a CICS administrator with CEDA access to the region).

```
ALTER TCPIPS(<name>) G(<groupName>) URM(XMLPCAN)
```

In order to create a new TCPIPS definition the following command is used.

```
DEFINE TCPIPS(<name>) G(<groupName>) URM(XMLPCAN)
PORTNUMBER(nnnn) PROTOCOL() - IIOP, HTTP, ECI
TRANSACTION(CWXN) TSQPREFIX(xxxxxx)
```

SSL() AUTHENTICATE()

Note: This customization is optional and if your installation doesn't enforce that transactions must run with appropriate SAF Id, it is not needed.

TCPIPS DEFINITION

If you're planning to run SOLA CICS Container with CICS TS 3.1 or above, there's a modification to the TCPIPS definition that you'll need to make to allow SOLA to communicate properly with CICS. Use the CEDA transaction (or equivalent) to update the field MAXDATALEN to a value greater than or equal to 4096 (the number is specified in kilobytes). MAXDATALEN specifies the maximum length of data that may be received by the TCP/IP service, and SOLA 6.1 increases the size of the messages that it can communicate.

Here is an example of a MAXDATALEN set to 5032 kilobytes.

```
I TCPIPS
  STATUS: RESULTS - OVERTYPE TO MODIFY
    Tcpips(TOREXT ) Ope Por(01743) Http Ssl Tra(CWXN)
      Con(00000) Bac( 00005 ) Max( 005032 ) Urm( XMLPCAN ) Sup
```

RACF AUTHORIZING THE SOLA TRANSACTIONS (OPTIONAL)

If your environment requires transactions to run authorized by RACF (or similar facility), and those transactions must be run under the RACF ID of the user, then you will need to use the XMLPCAN analyzer, as defined in section "Define the CICS Analyzer to use with SOLA (Optional)" on page 34. This will run SOLA's main transaction (by default this is transaction XML) under the RACF ID that was extracted from the WS-Security header by XMLPCAN.

This is the list of transactions used by SOLA that will need to be RACF authorized:

- XML: A requestor's RACF ID will need to be authorized to run this transaction.
- XML3: SOLA's runtime UDDI interface. Because of the open nature of the UDDI interface this transaction can't run under the RACF ID of the requestor.
- XML8: SOLA's background task to spool metrics information from the CICS metrics UMT to a DB2 table.
- XML5, XML7 and XML9: These transactions are used when SOAP requests and responses are carried by MQ.

DEFINING IMS COMPONENTS (OPTIONAL)

SOLA CICS Container includes the optional ability to invoke IMS-TM transactions and subroutines as reusable web services. You will need to use your standard IMS procedures to define the transaction and PSB that SOLA uses to invoke IMS

subroutines. No IMS transactions or PSBs are required to use SOLA with IMS transactions, but they are required if you wish to use the sample transactions. The IMS macro instructions are contained in members IMMSGEN (to define the IMS subroutine driver and sample transactions) and PSBGEN (to define the PSBs for the subroutine driver programs and the define the sample program). The definitions for the sample program and transaction are required for installation verification.

SETTING UP THE IMS MESSAGE PROCESSING REGION (OPTIONAL)

Before you can use SOLA to invoke IMS transactions and subroutines you will need to add the SOLA Runtime Library to the STEPLIB concatenation of the IMS Message Processing Region (MPR).

POPULATING DATA FOR SAMPLE APPLICATION (RECOMMENDED)

The SOLA CICS Container package includes a sample application that is used to debug and verify the installation. In order to run this application the SAMPLIB contains two members that are used to populate the DB2 tables used by the sample application. The tables are created as a part of the customization. Members INSWGT and INSRTUAP in the SAMPLIB dataset contain the SQL to insert this data and it can be done using SPUFI or any other facility.

MIGRATION TO SOLA6.1 FROM SOLA5.1 (OPTIONAL)

Customers who already have SOLA5.1 should refer to the SOLA 6.1 Migration Guide, which documents the following process for data migration. The steps involved are:

- Customize the migration jobs, etc with SOLAEDT
- Run the job DDLJCLM to create alias entries for SOLA5.1 tables. This job requires DB2 database privileges and may need to be run by a DBA.
- Bind the SOLA Migration Program packages by running job BINDJCLM
- Run the migration Jobs in the following sequence**

Job	Description	MAX RC
MIGJOB1	Verify SOLA5.1 Directory to check if there are any potential migration issues	4
MIGJOB2	Prepare SOLA6.1 Directory for a clean migration from SOLA5.1 Directory	4
MIGJOB3	Migrates SOLA Directory – Project, Program and	4

	related data	
MIGJOB4	Migrates SOLA Directory – User, Access related data	4
MIGJOB5	Migrates SOLA Directory – All other data	4

** The migration process can be repeated by repeating the above jobs starting from MIGJOB1

If you are a SOLA 6.0 test customer and you've already installed the SOLA 6.1 Server, then you will need to restart the SOLA 6.1 Server after the migration batch jobs have completed.

Chapter 3: Customizing SOLA IMS Container on a z/Series mainframe (SOLS600)

CUSTOMIZING THE SOLA INSTALLATION

The following sections describe the steps necessary to customize the sample members and jobs in the *<tlibhlq>.SAMPLIB* and then run those jobs to:

- Customize the SAMPLIB
- Run jobs in SAMPLIB to:
 - Create the SOLA DB2 database
 - Bind the plan and all of the packages in the SOLA collection

CUSTOMIZING THE *<TLIBHLQ>.SAMPLIB*

Refer to the #README member of *<tlibhlq>.SAMPLIB*.

Run the SOLAEDT macro on the following members of *<tlibhlq>.SAMPLIB*:

Member	Description
BIND, BIND1	Plan and Package bind commands
BINDJCL	JCL to bind the plan and packages
DCL	DB2 Grant statements
DDL	DDL to create the DB2 STOGROUP (optional), database, tablespaces, tables, indexes and aliases.
DDLJCL	JCL to create the DB2 entries
IMSGEN	Macro instructions to define the IMS subroutine driver and sample transactions.
PSBGEN	Macro instructions to define the PSBs for the subroutine driver programs and the sample program.

Member	Description
SOLA	This is a SOLA HTTP Started Task Job Sample. Customize this JCL and copy to your started task library
SOLAPRM	This is a SOLA HTTP Started Task Job parms Customize this parmlib and copy to your parm library
SOLAMQ	This is a SOLA MQ Started Task Job Sample. Customize this JCL and copy to your started task library
SOLAMQPR	This is a SOLA HTTP Started Task Job parms Customize this parmlib and copy to your parm library

CREATING THE DATABASE WITH DDLJCL

You create the SOLA directory by running the job DDLJCL. This job requires DB2 database privileges and may need to be run by a DBA. In order to successfully run the job you must have appropriate DB2 authority to create a STOGROUP (optional), a DATABASE (DBXML002), TABLESPACE, TABLE, INDEX and ALIAS.

The SOLA directory is defined in a single database DBXML002. You can create this database in its own STOGROUP or you can use an existing STOGROUP. If you want to use a STOGROUP specifically for DBXML002 then uncomment the "CREATE STOGROUP" statement at the beginning of member DDL.

The SOLA Database DBXML002 consists of at least 28 tables. Each table is created in its own unique tablespace. There are at least 47 indexes. At least 28 ALIASes are defined.

Job DDLJCL should execute with a return code of zero.

BINDING THE PLAN AND PACKAGES WITH BINDJCL

You bind the SOLA packages and plan by running jobs BINDJCL.

The person running these jobs must have appropriate authority to select, update, insert and delete the tables in the DBXML002 database.

There are at least 28 package bind cards in member BIND. All of these packages are bound into a single collection <collection>. There is a single plan bind statement for plan <plan> in member BIND1 which includes all packages in PKLIST <collection>.

Job BINDJCL should execute with a return code of zero.

RACF AUTHORIZATION OF SOLA IMS CONTAINER

In accordance with your site security requirements, define the <SOLAStartedTaskName> to RACF 'Started' Class to assign a STC userid under which the job must run.

Make sure that the STC Userid you assign:

- Has OMVS UID defined
- Has access to SOLA libraries
- Granted EXECUTE on SOLA DB2Plan (Refer to member DCL in samplib).

APF AUTHORIZE THE SOLA LOAD LIBRARY (RECOMMENDED)

To allow the SOLA IMS Container to validate Security credentials passed through a SOAP request, the SOLA Loadlibrary must be APF authorized. Your z/OS Administrator can do this by adding the <tlibhlq>.LOADLIB to your system parmlib PROGxx and dynamically activating the parm with SET PROG=xx.

DEFINING IMS COMPONENTS

The SOLA IMS Container includes the ability to invoke IMS-TM transactions and subroutines as reusable web services. You will need to use your standard IMS procedures to define the transaction and PSB that SOLA uses to invoke IMS subroutines. No IMS transactions or PSBs are required to use SOLA with IMS transactions, but they are required if you wish to use the sample transactions. The IMS macro instructions are contained in members IMMSGEN (to define the IMS subroutine driver and sample transactions) and PSBGEN (to define the PSBs for the subroutine driver programs and define the sample program). The definitions for the sample program and transaction are required for installation verification.

SETTING UP THE IMS MESSAGE PROCESSING REGION (OPTIONAL)

Before you can use SOLA to invoke IMS transactions and subroutines you will need to add the SOLA Runtime Library to the STEPLIB concatenation of the IMS Message Processing Region (MPR).

CONFIGURING THE SOLA IMS CONTAINER

The SOLA IMS Container runs as a z/OS started task. The Started task can be setup as a HTTP or MQ based SOLA server. It incorporates many of the features of the SOLA CICS Container, but because it runs without CICS there are several parameters that need to be specified to make it work.

The parameters are specified in a PARMLIB dataset that's made available to the Started Task by specifying it in a DD statement in the Started Task JCL.

Started Task JCL

The following sample JCL is provided in the SOLA SAMPLIB. Customize this JCL to conform to your installation requirements. The JCL can be customized using the SOLAEDT Rexx Edit Macro that you customized during the installation of SOLA.

SOLA STC Proclib

```
/* SOLA STARTED TASK
//LISTEN EXEC PGM=XMLPC125,
//           REGION=<RegionSize>, DYNAMNBR=20
//STEPLIB  DD DISP=SHR, DSN=<tlibh1q>.LOADLIB
//           DD DISP=SHR, DSN=<DB2.SDSNLOAD>
//           DD DISP=SHR, DSN=<Db2.SDSNEXIT>
//SOLALIB  DD DISP=SHR, DSN=<tlibh1q>.LOADLIB
//           DD DISP=SHR, DSN=<Application Template Library>
//SYSUT1   DD DSN=&UT1,
//           SPACE=(1700,(400,50)),
//           UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
///*DSNTRACE DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*, SPIN=UNALLOC, FREE=CLOSE
//PARMLIB  DD DISP=SHR, DSN=<Parmlib(SOLAPRMS)>
```

<RegionSize>

The region size that's required to run the SOLA STC can be computed as follows:

Base Region size needed = 3 MB
Each SOLA Thread will consume = 0.5 MB

If you setup a SOLA Started task to process a maximum of 50 threads then the region size required is 3 MB + (50*.0.5 MB) = 28 MB

<Application Template Library(Libraries)>

Concatenate your application template library(libraries) to the SOLALIB DD card.
These datasets are load libaries where generated SOLA template loadmodule artefacts
are stored

<Parmlib(SOLAPRMS)> (2

```
DTYP=HTTP | MQ
TRCE=0
SYST=SOL1
PLAN=XMLCLNT
DB2S=DB9G
PROT=000
MAXT=200
FCTM=02000
TIME=00100
MTSQ=09999
IDLE=86400

** For HTTP Server **
PORT=01449
TCPIP

** For MQ Server **
QMGR=<Qmgr>
REQQ=<MQRequestQ>
```

Specify each parameter exactly as shown. Each parameter must start on a separate line and begin in column 1. Don't abbreviate numeric fields; each field must have the requisite number of digits, as specified below.

SYST: Mandatory

4 alphanumeric characters. The SOLA System ID (equivalent of CICS SYSID) assigned to the SOLA Started Task instance.

PORT: Mandatory (** For HTTP Server**)

5 numeric digits. SOLA Started Task Listener TCPIP Port Number

DB2S: Mandatory

4 alphanumeric characters. DB2 Subsystem in which the SOLA directory is created

PLAN: Mandatory

8 alphanumeric characters. DB2 Plan to be used

TRCE: Optional

1 numeric digit. Sets the trace Level for debugging the SOLA Started Task.

Values : 0 – 9.

Default: 0 (No Trace)

PROT: Optional

3 numeric digits. Specifies the number of protected threads to be created.

Default: 000 (No protected threads created at start-up of SOLA STC instance)

MAXT: Optional

3 numeric digits. Specifies the maximum number of concurrent threads to be supported by the SOLA STC instance

TCPN: Optional (**For HTTP Server**)

8 alphanumeric characters. TCPIP address space name on the system that SOLA will connect to.

Default: TCPIP

FCTM: Optional

5 numeric digits. Specifies the number of SOLA Internal Logging File Control records to be handled by the SOLA STC instance.

Default: 02000

MTSQ: Optional

5 numeric digits. SOLA STC caches runtime metadata into internal memory areas called TSQs. This parameter defines the number of TSQs to be supported by the specific SOLA STC instance.

Default: 09999

TIME: Optional

5 numeric digits. This is an SOLA internal control parameter that indicates how long in milliseconds that the listener should wait before the incoming socket connection is taken by a subtask.

Default: 00100

IDLE: Optional

5 numeric digits. SOLA STC thread manager uses the value specified in this parm to control when an IDLE thread is to be released. The value specified in this parm indicates number of seconds after which an Inactive(Idle) SOLA thread needs to be terminated

Default: 86400

QMGR: Mandatory (**For MQ Server**)

4 alphanumeric characters. MQ Subsystem to which the SOLA MQ server connects.
Please note that CSD Definition and XML# definition for TRANCLASS is shipped with the default DFHTCL00 and if MQ is going to be used, must be customized at setup.

REQQ: Mandatory (**For MQ Server**)

Max 48 character SOLA Request Queue Name. This is the Queue to which applications route the soap requests to be processed by SOLA MQ Server. *Please note that CSD Definition and XML# definition for TRANCLASS is shipped with the default DFHTCL00 and must be customized at setup.*

Chapter 4: Installing the SOLA Development Studio

OVERVIEW

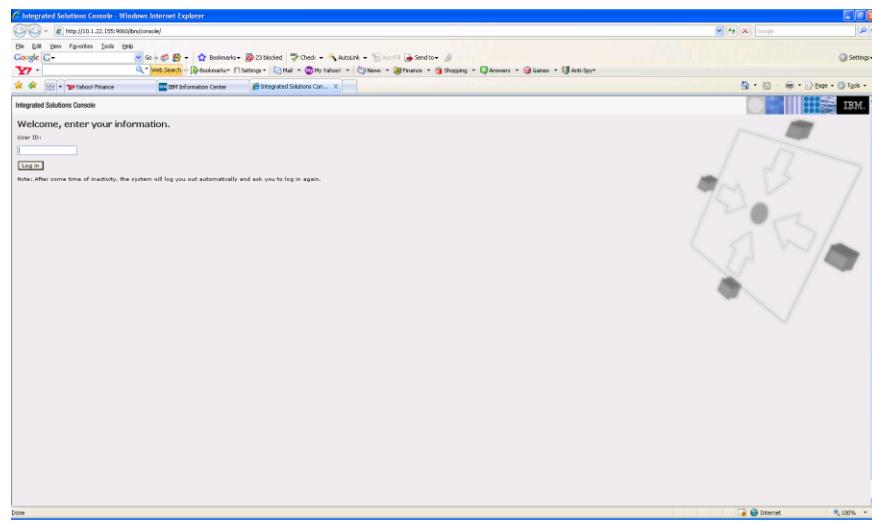
The SOLA Installation includes host application server installation components. The installation package contains both an .ear file and a .war file. You start by choosing whether to install from the .ear or the .war, then you begin by copying the installation .ear or .war file to an install directory.

SOLA can be installed on WebSphere, WebLogic and Tomcat. Please follow the instructions in the appropriate chapter for your environment.

INSTALLING THE SOLA DEVELOPMENT STUDIO IN WEBSPHERE

This section provides instructions for installing the SOLA Development Studio into a WebSphere Application Server environment. The SOLA Development Studio requires WebSphere WAS 6.1 or greater. In order to begin installation of the SOLA Development Studio, you must have administrator privileges on the WAS computer that you're doing the installation on.

Installing SOLA in WebSphere

Step	Procedure
1.	<p>Log on to WebSphere's Admin console.</p>  <p>Figure 5: WebSphere Application Server –Logon page</p>

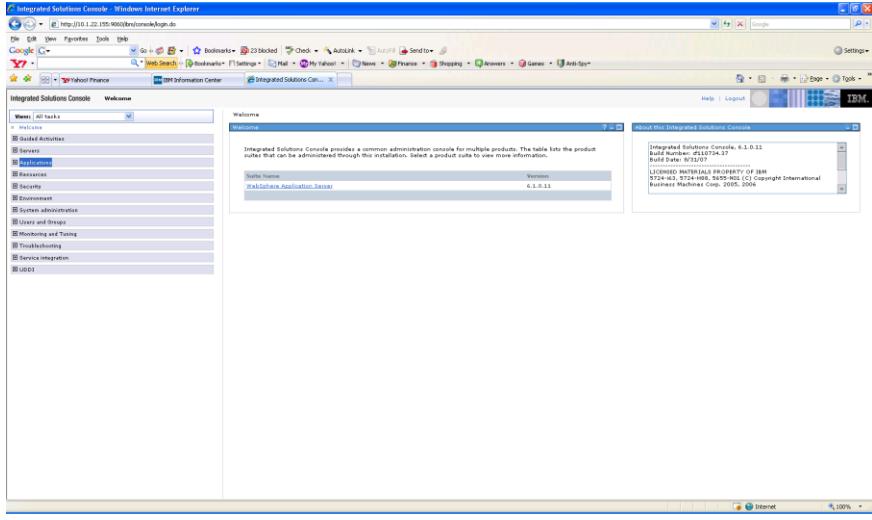
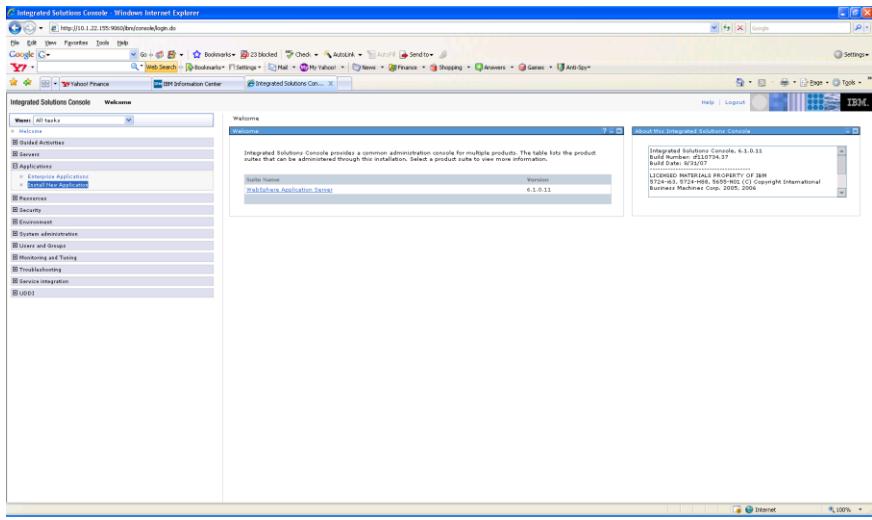
Step	Procedure
2.	<p>Click “Applications” within navigation area to expand link.</p> 
3.	<p>Click “Install New Application” within navigation area.</p> 

Figure 6: WebSphere Application Server –Integrated Solutions Console

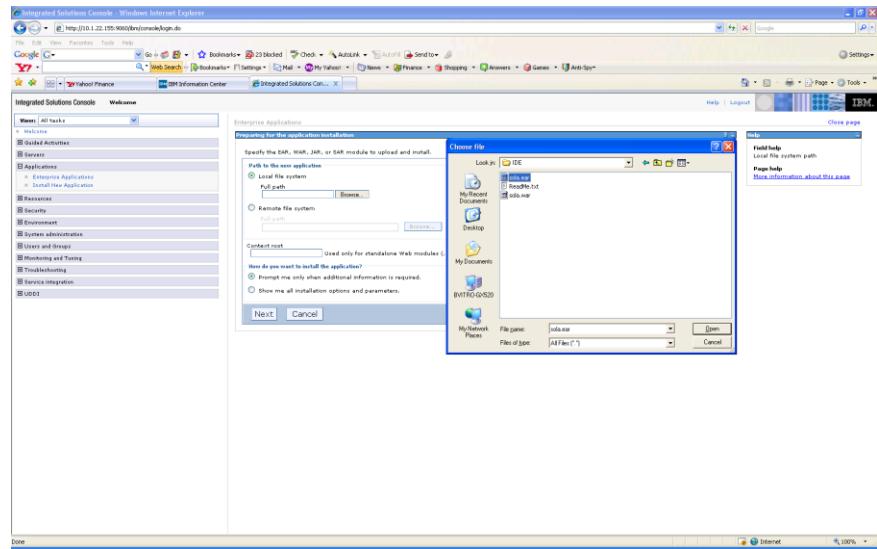
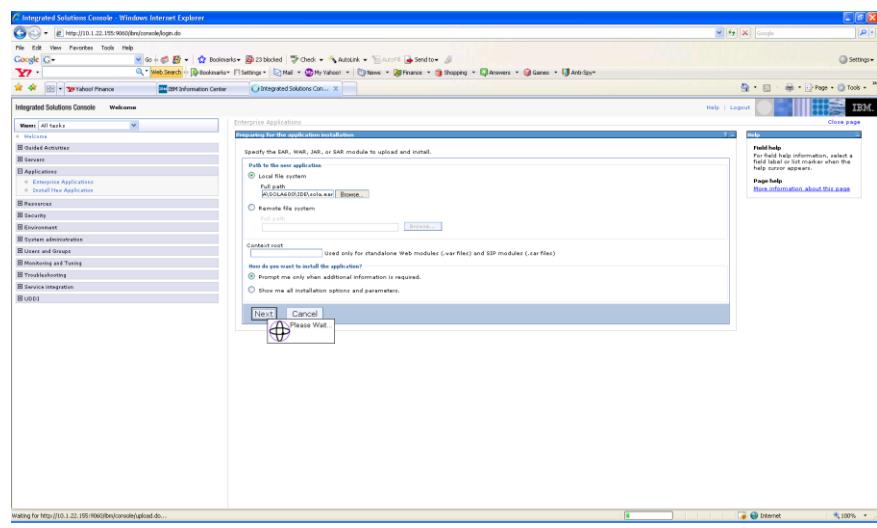
Step	Procedure
4.	<p>Click “Browse...” button within content area to select the location of your sola.ear application file.</p> 
5.	<p>Click the “Next” button within content area. This step uploads the sola application ear file to the server and may take several minutes to complete.</p> 

Figure 8: WebSphere Application Server –Select the sola.ear file**Figure 9: WebSphere Application Server –Upload the sola.ear file**

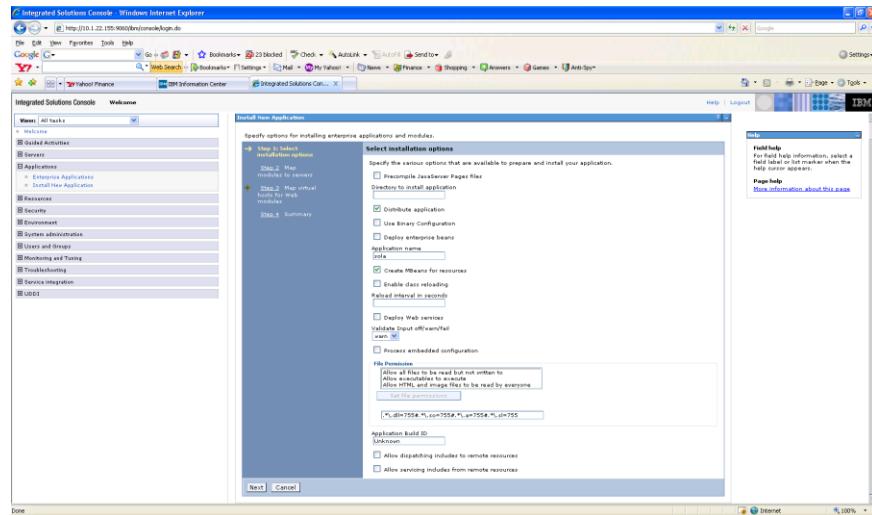
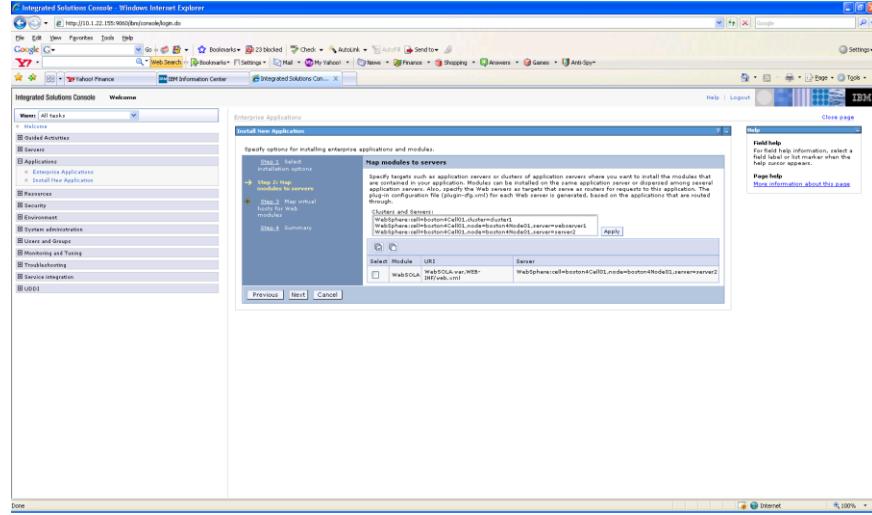
Step	Procedure
6.	<p>Make any choices specific to your installation or Click "Next" to continue with default settings.</p> 
7.	<p>Make any choices specific to your installation or Click "Next" to continue with default settings.</p> 

Figure 10: WebSphere Application Server –Select Installation Options

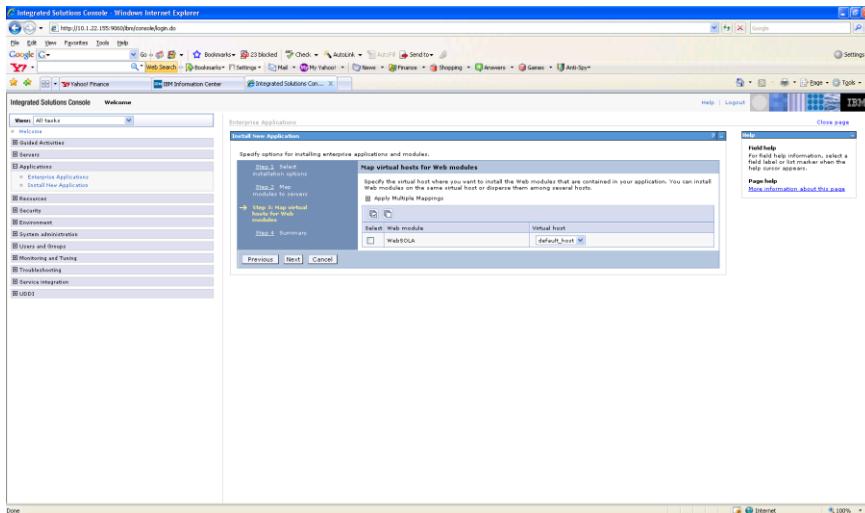
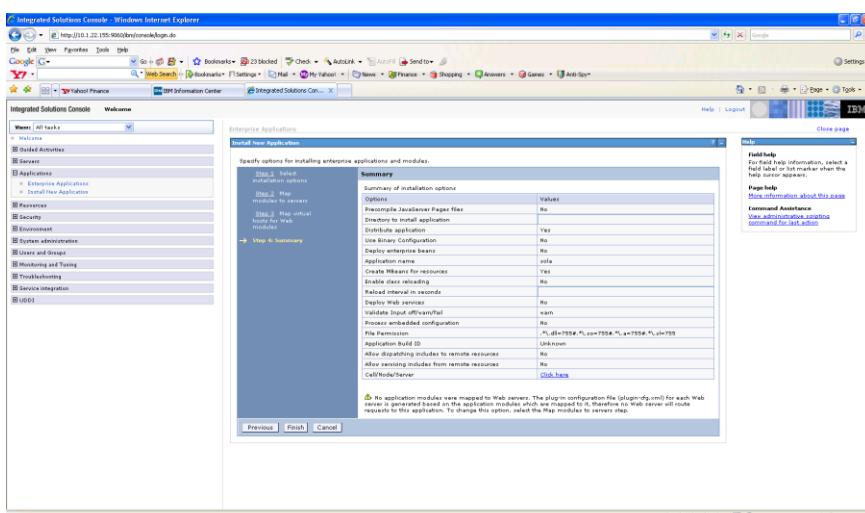
Step	Procedure
8.	<p>Make any choices specific to your installation or Click "Next" to continue with default settings.</p> 
9.	<p>Make any choices specific to your installation and Click "Finish" to complete the application installation.</p> 

Figure 12: WebSphere Application Server –Map virtual hosts for web modules

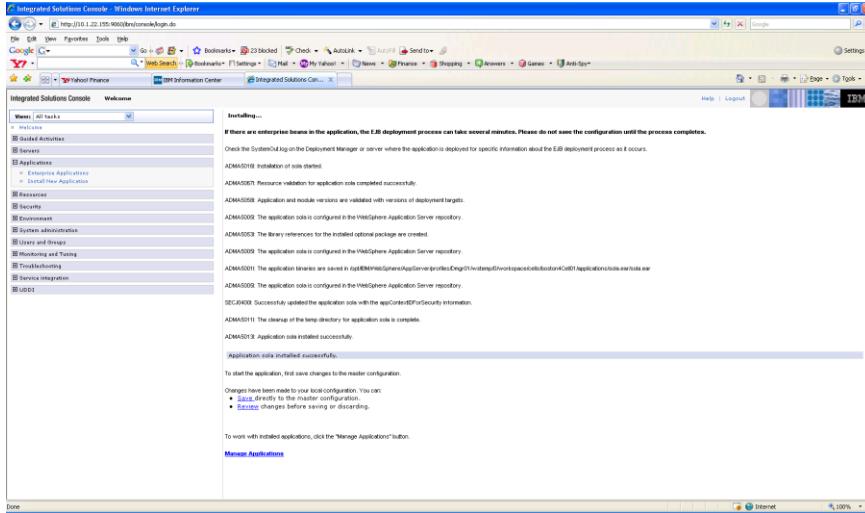
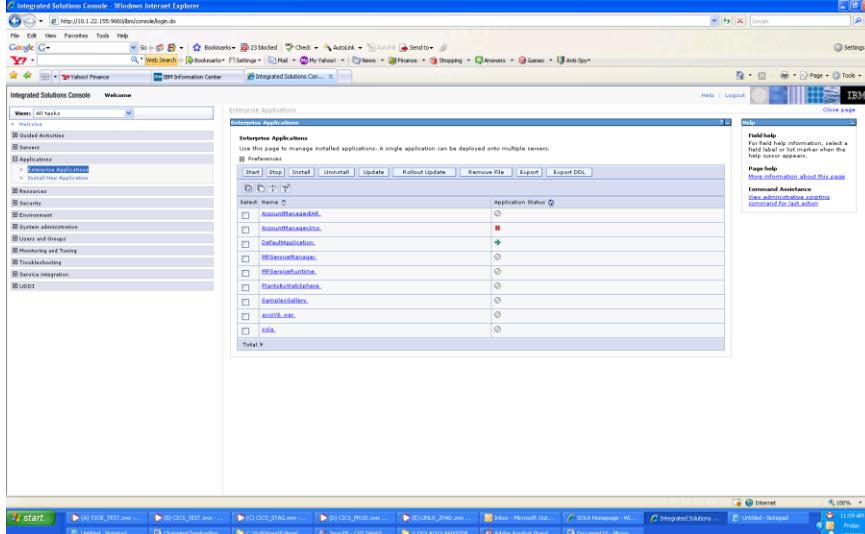
Step	Procedure
10.	<p>Click “Save” link within the content area to finalize the installation.</p> 
11.	<p>Click “Enterprise Application” to see the application you just installed. Unless you explicitly changed it the SOLA application will be named “sola”. Click on that link to configure the class loading policy that is required when using WebSphere Application Server.</p> 

Figure 14: WebSphere Application Server –Save

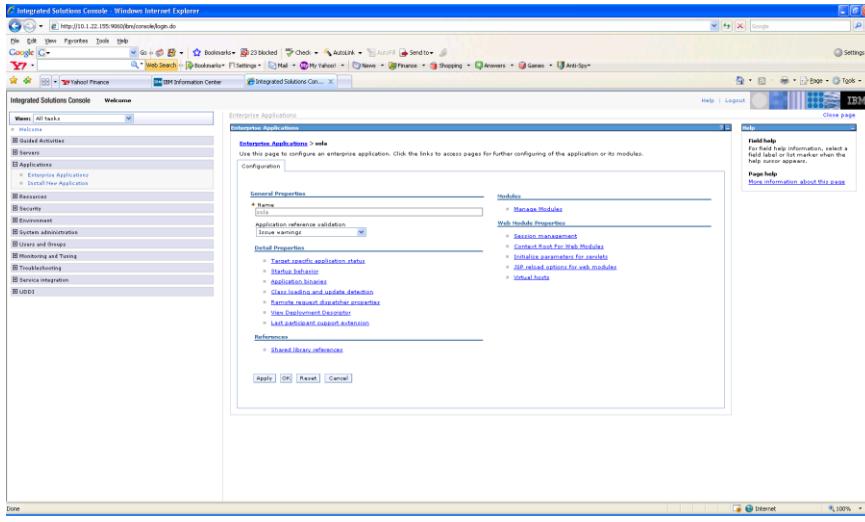
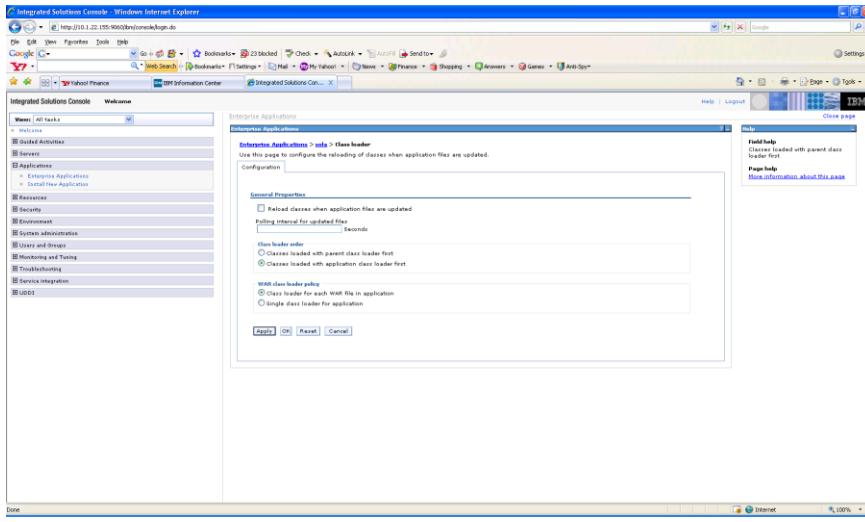
Step	Procedure
12.	<p>Click the “Class loading and update detection” link within the content area.</p> 
13.	<p>Select “Classes loaded with application class loader first” and “Class loader for each WAR file in application” and click the “Ok” button.</p> 

Figure 16: WebSphere Application Server –Class loading**Figure 17: WebSphere Application Server –General Properties**

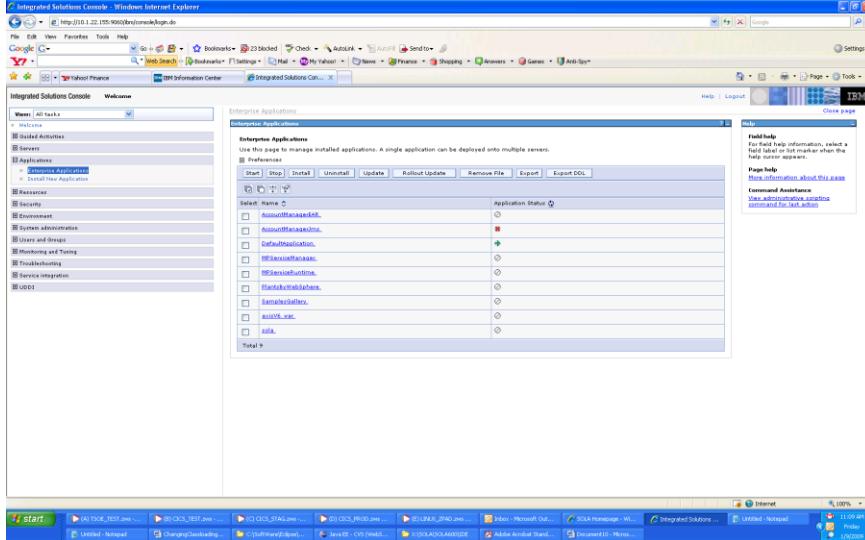
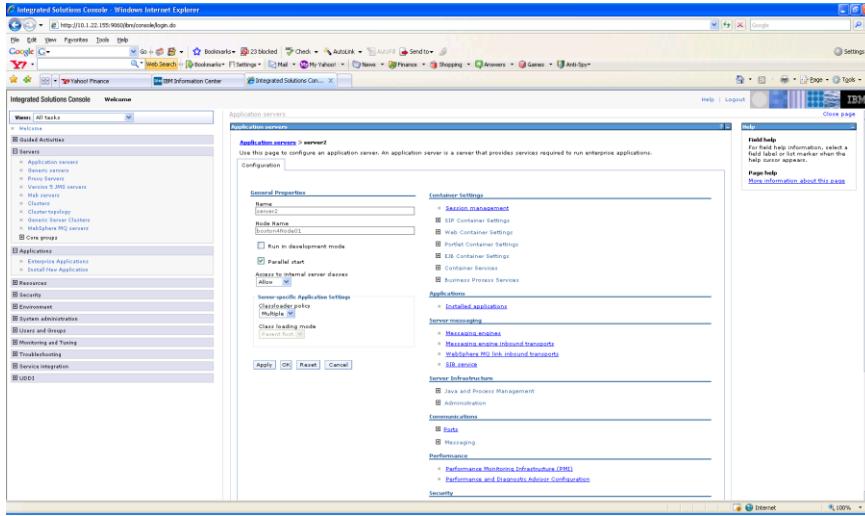
Step	Procedure
14.	<p>Click “Enterprise Application” to see the application you just installed. Unless you explicitly changed it the SOLA application will be named “sola”. Click on that link to configure the location of the “SOLARoot” directory where SOLA stores customizable information..</p> 
15.	<p>Click “Java and Process Management” to expand that link.</p> 

Figure 18: WebSphere Application Server –Configuring the SOLARoot

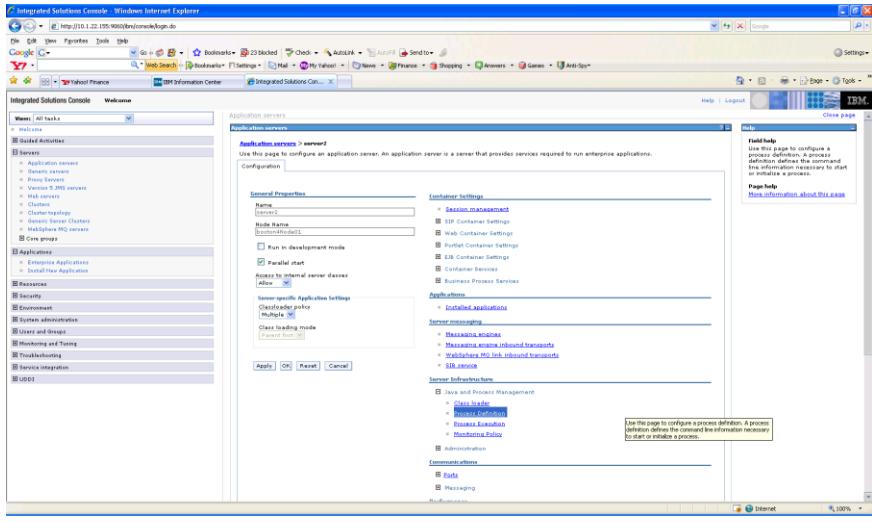
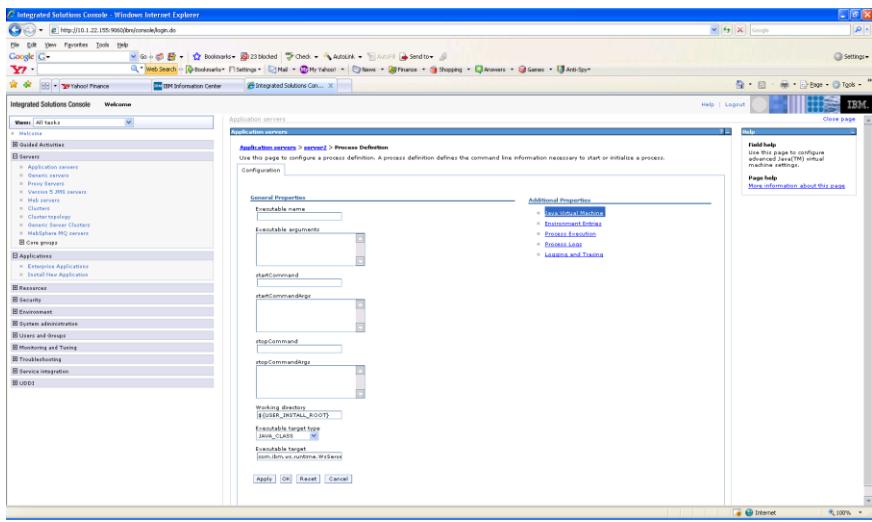
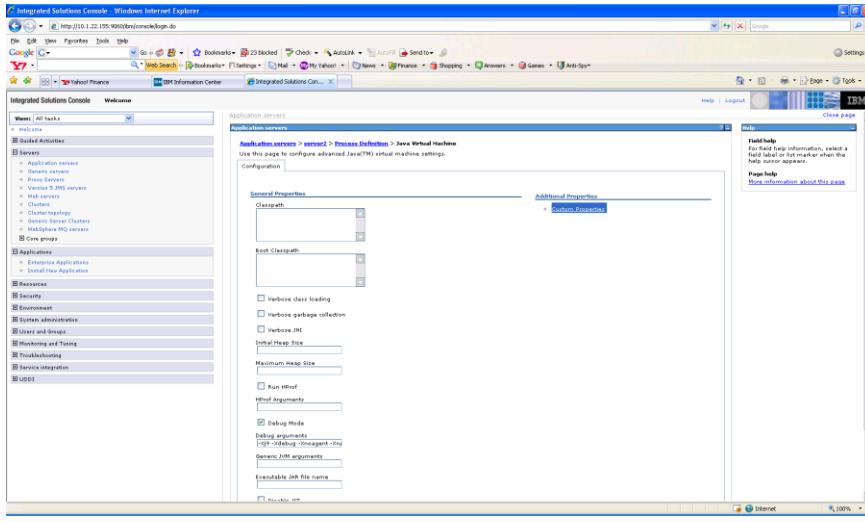
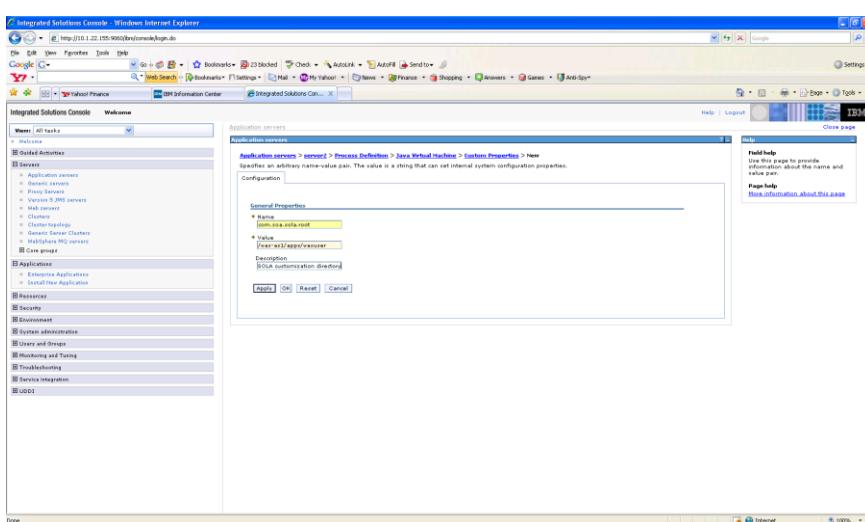
Step	Procedure
16.	<p>Click on the “Process Definition” link.</p> 
17.	<p>Next click on the “Java Virtual Machine” link which will allow you to enter the root directory which will contain SOLA’s configuration info.</p> 

Figure 20: WebSphere Application Server –Process Definition

17. Next click on the “Java Virtual Machine” link which will allow you to enter the root directory which will contain SOLA’s configuration info.

Figure 21: WebSphere Application Server –JVM

Step	Procedure
18.	<p>Click on the “Custom Properties” link</p> 
19.	<p>The content section now contains a panel which will allow you to enter a system property (which represents SOLA’s root directory to contain configuration info). After you set the property named “com.soa.sola.root” to the root location of where you want your config data to be stored click “Ok”.</p> <p>In the example below, “/was-as1/apps/wasuser” is the file system which SOLA will use. You can create a new directory for use by SOLA.</p> <p>Note: The authority of the JVM must have read/write/execute access on this directory or folder.</p> 
20.	Proceed to chapter 4 for customization.

INSTALLING THE SOLA DEVELOPMENT STUDIO IN WEBLOGIC

This section provides instructions for installing the SOLA Development Studio into a WebLogic Application Server environment. The SOLA Development Studio requires BEA WebLogic Server 8.1 or greater. In order to begin installation of the SOLA Development Studio, you must have administrator privileges on the WebLogic computer that you're doing the installation on.

Installing SOLA in WebLogic

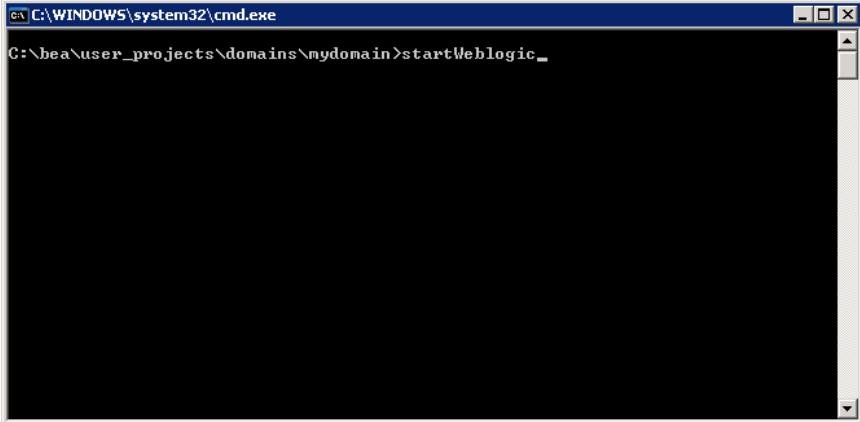
Step	Procedure
1.	Point to the SOLA installation package. Change to the /ide/ear/ directory.
2.	Start the Weblogic Server by invoking the startup script as shown below. 

Figure 24: WebLogic –Start-up

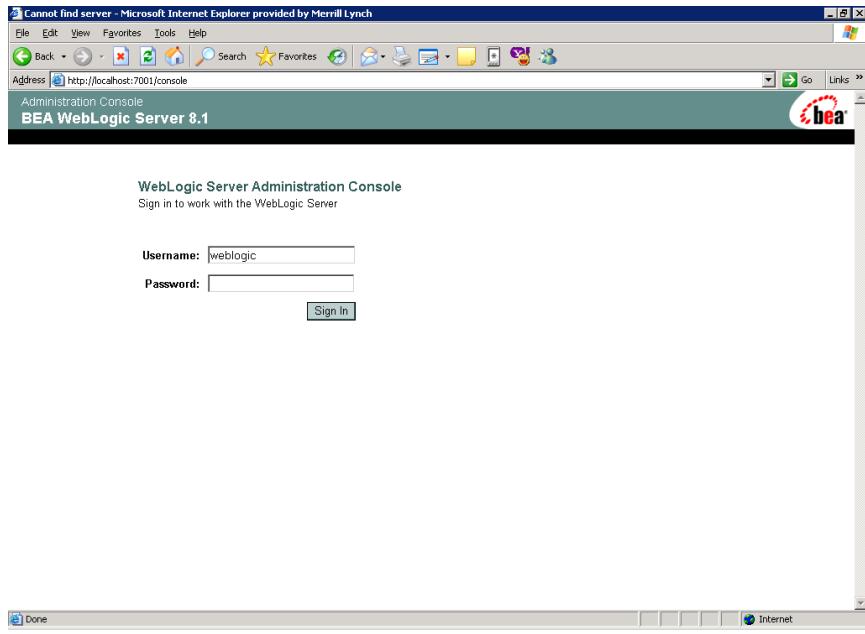
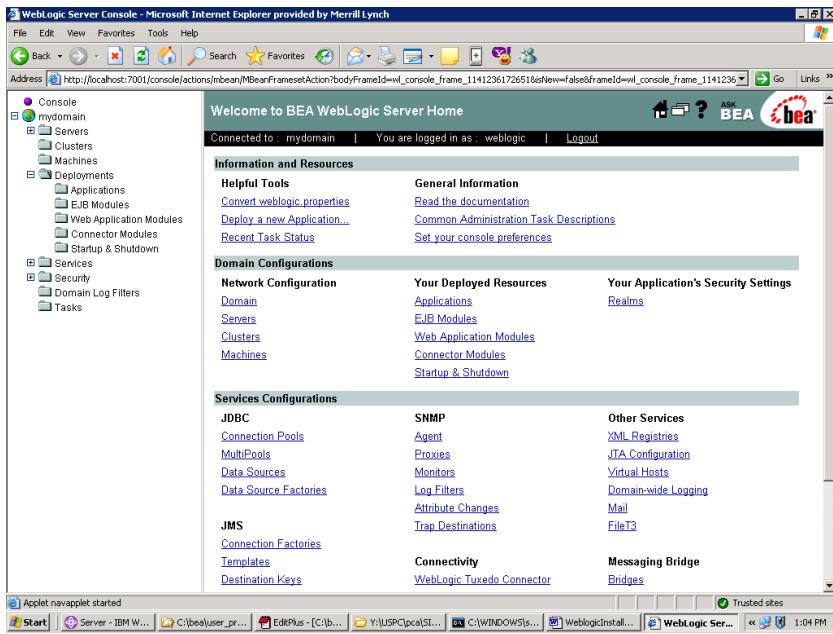
Step	Procedure
3.	<p>Once the server has started successfully, load the Administration console screen using the following URL: e.g. - http://localhost:7001/console</p> 
4.	<p>Sign in. A successful sign in will navigate the user to the screen shown below</p> 

Figure 25: WebLogic Server –Administration Console

Figure 26: WebLogic –Server Home

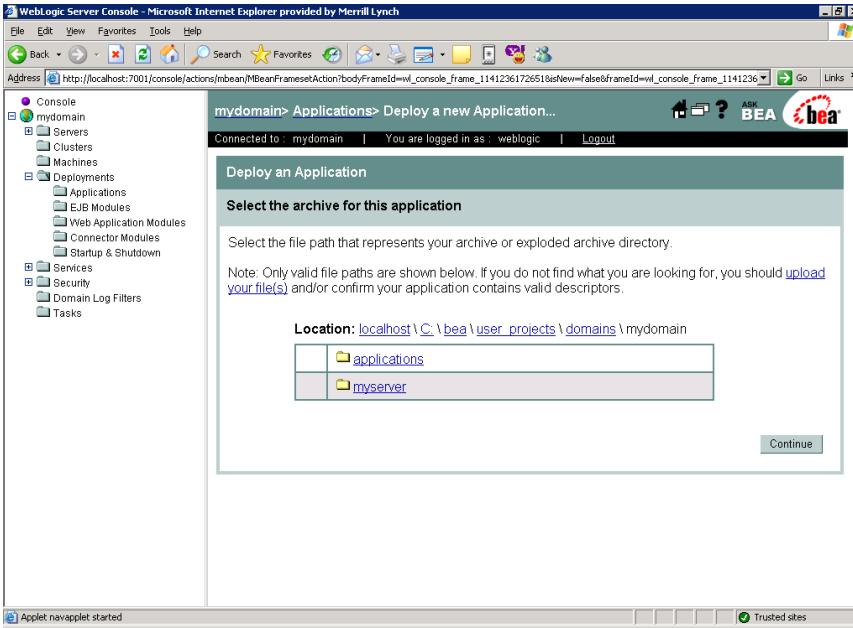
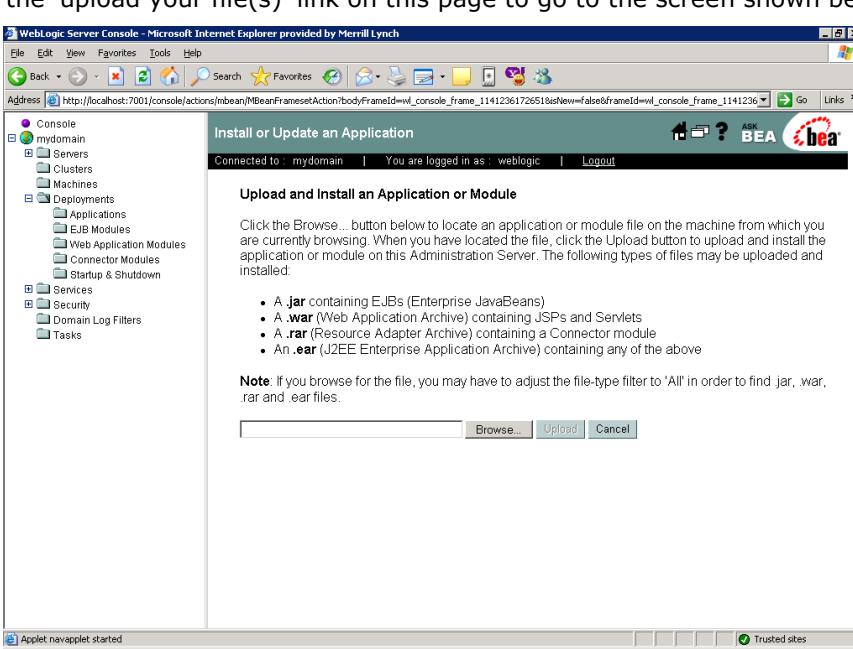
Step	Procedure
5.	<p>Click on 'Deploy a new Application' link on this screen to go to the screen shown below.</p> 
6.	<p>Click on the 'upload your file(s)' link on this page to go to the screen shown below.</p> 

Figure 27: WebLogic –Deploy a new application

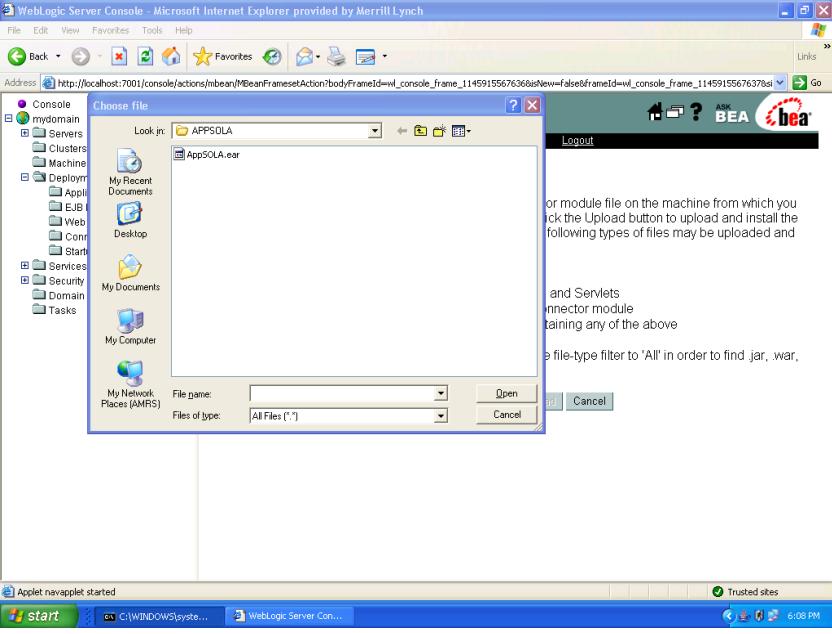
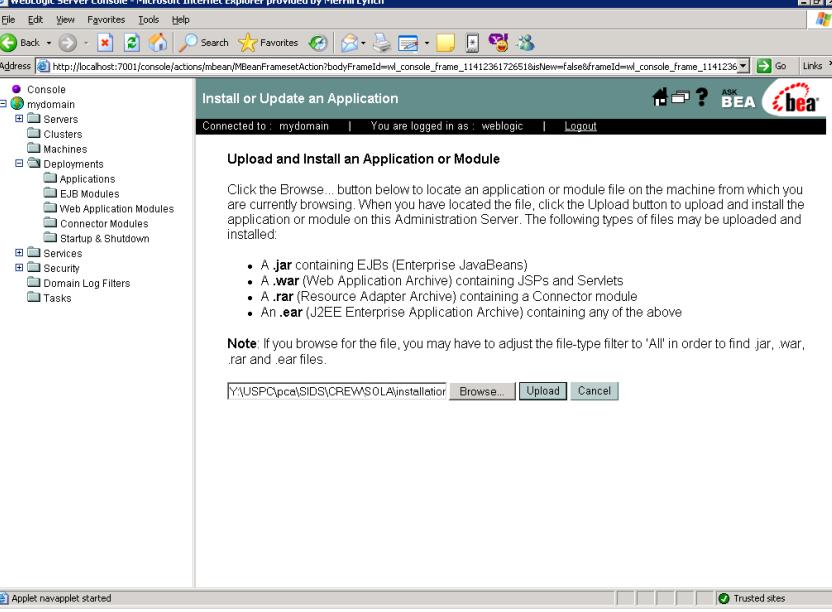
Step	Procedure
7.	<p>Click on Browse to select the .ear file. Select the .ear file in the 'Choose file' window and click on the 'Open' button.</p> 
8.	<p>Once the file is selected, click on the 'Upload' button to upload the 'ear' file to the Weblogic server.</p> 

Figure 29: WebLogic –Choose file

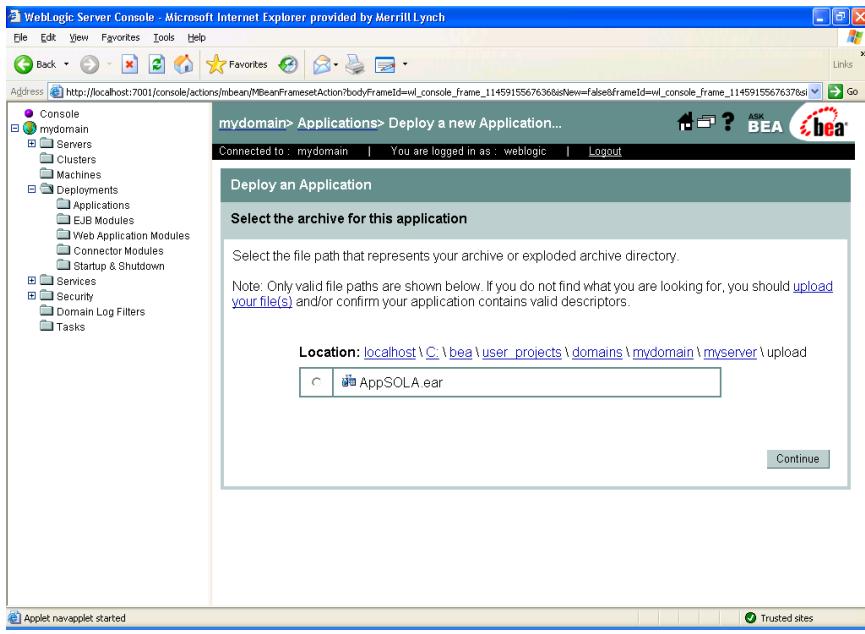
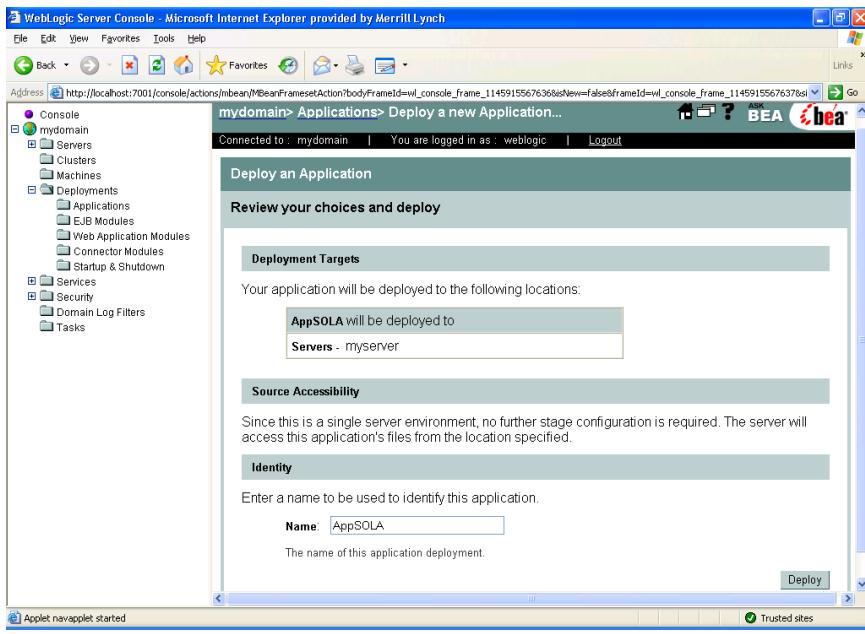
Step	Procedure
9.	<p>Check the radio button on this screen against the 'AppSOLA.ear' file name and click on the 'Continue' button.</p> 
10.	<p>Click on the 'Deploy' button on this screen.</p> 

Figure 31: WebLogic –Deploy an Application

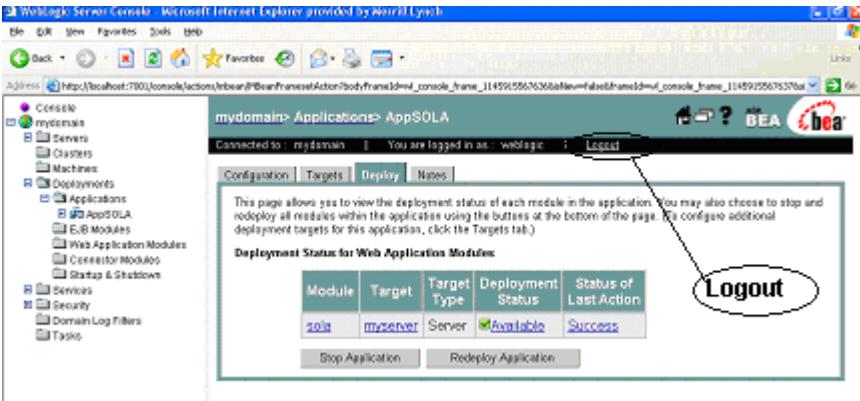
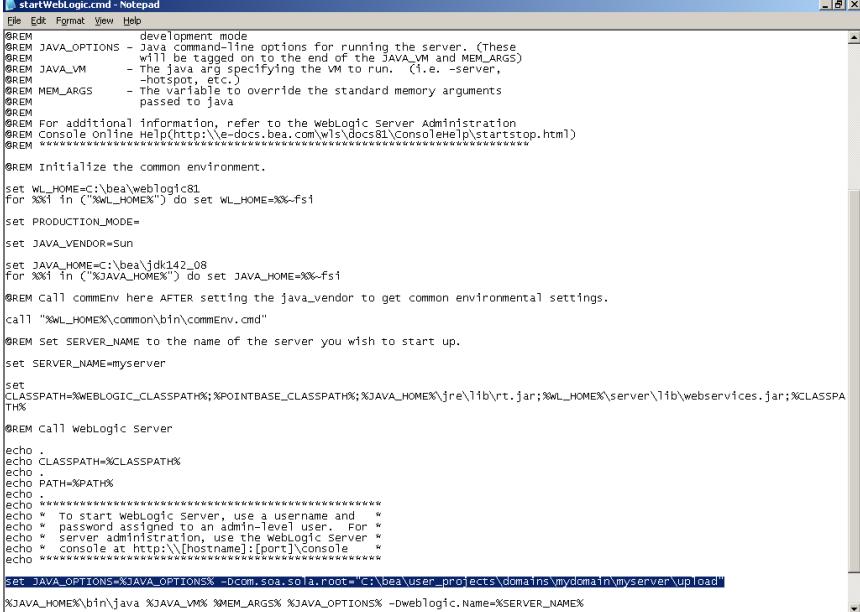
Step	Procedure
11.	<p>On successful deployment, the Status shows as 'Success', as shown below. The application is deployed now. Logout of this screen.</p>  <p>The screenshot shows the WebLogic Server Console in Microsoft Internet Explorer. The URL is <code>http://localhost:7001/console/actions/tbserv/BearFrontendAction/Body/frameset_console_frame_114591586762686Name=false&frameId=mut_console_frame_11459158676370as</code>. The left sidebar shows the domain structure: mydomain > Applications > APPSOLA. The main content area displays the 'Deployment Status for Web Application Modules' table. The table has columns: Module, Target, Target Type, Deployment Status, and Status of Last Action. One row is visible: Module is 'soa', Target is 'myserver', Target Type is 'Server', Deployment Status is 'Available', and Status of Last Action is 'Success'. At the bottom of the table are 'Stop Application' and 'Redeploy Application' buttons. A red oval highlights the 'Logout' link in the top right corner of the main content area.</p>
12.	<p>On the StartWeblogic script (you can find it in domain directory where Web logic is installed), add a new line of code as shown below:</p> <pre>set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.soa.sola.root=<fileSystem></pre> <p>Where <fileSystem> is the directory where SOLA will store all its files.</p> <p>For example: if you have created a new directory "C:\SOLAFiles", you would add the following code.</p> <pre>set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.soa.sola.root="C:\SOLAFiles"</pre> <p>Save and close this file</p>  <p>The screenshot shows a Windows Notepad window titled 'startWeblogic.cmd - Notepad'. The content of the script is as follows:</p> <pre> REM development mode REM JAVA_OPTIONS - Java command-line options for running the server. (These REM will be tagged on to the end of the JAVA_VM and MEM_ARGS) REM JAVA_VM - The variable specifying the VM to run. (i.e. -server, REM -hotspot, etc.) REM MEM_ARGS - The variable to override the standard memory arguments REM passed to java REM REM For additional information, refer to the webLogic Server Administration REM Console Online Help(http://e-docs.bea.com/wldocs81/consoleHelp/startstop.html) REM ***** REM Initialize the common environment. set WL_HOME=:bea\weblogic81 for %%I in ("%WL_HOME%") do set WL_HOME=%%I-fsi set PRODUCTION_MODE= set JAVA_VENDOR=sun set JAVA_HOME=c:\bea\jdk142_08 for %%I in ("%JAVA_HOME%") do set JAVA_HOME=%%I-fsi REM Call commenv here AFTER setting the java_vendor to get common environmental settings. call "%WL_HOME%\common\bin\commEnv.cmd" REM Set SERVER_NAME to the name of the server you wish to start up. set SERVER_NAME=myserver set CLASSPATH=%WEBLOGIC_CLASSPATH%;%POINTBASE_CLASSPATH%;%JAVA_HOME%\jre\lib\rt.jar;%WL_HOME%\server\lib\webservices.jar;%CLASSPA TH% REM Call webLogic server echo . echo CLASSPATH=%CLASSPATH% echo . echo PATH=%PATH% echo ***** echo * To start weblogic server, use a username and * echo * password assigned to an admin-level user. For * echo * server administration, use the webLogic server * echo * console at http://[hostname]:[port]/console * echo ***** set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.soa.sola.root="C:\bea\user_projects\domains\mydomain\myserver\upload" %JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS% -DwebLogic.Name=%SERVER_NAME% </pre>

Figure 34: WebLogic –Start WebLogic Script

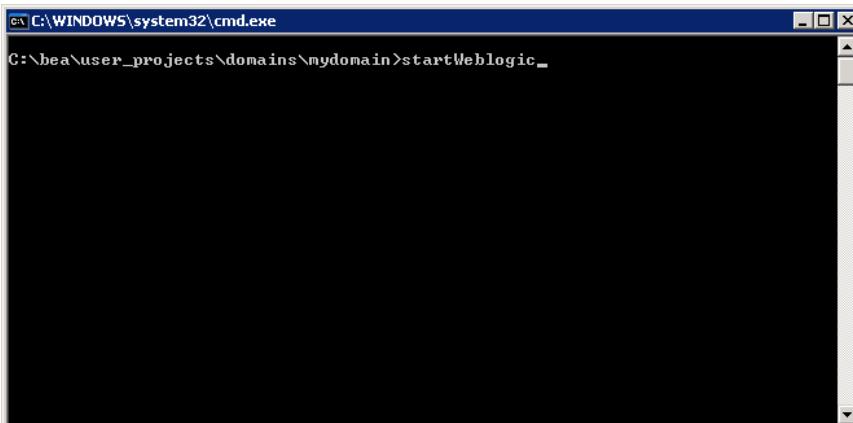
Step	Procedure
13.	<p>Restart the server now as shown below.</p> 
14.	<p>Proceed to chapter 4 for customization.</p>

Figure 35: WebLogic –Start-up

INSTALLING THE SOLA DEVELOPMENT STUDIO IN TOMCAT

This section provides instructions for installing the SOLA Development Studio into a Tomcat Application Server environment. The SOLA Development Studio requires Apache Tomcat 4.1 or greater. In order to begin installation of the SOLA Development Studio, you must have administrator privileges on the Tomcat computer that you're doing the installation on.

Installing SOLA in Tomcat

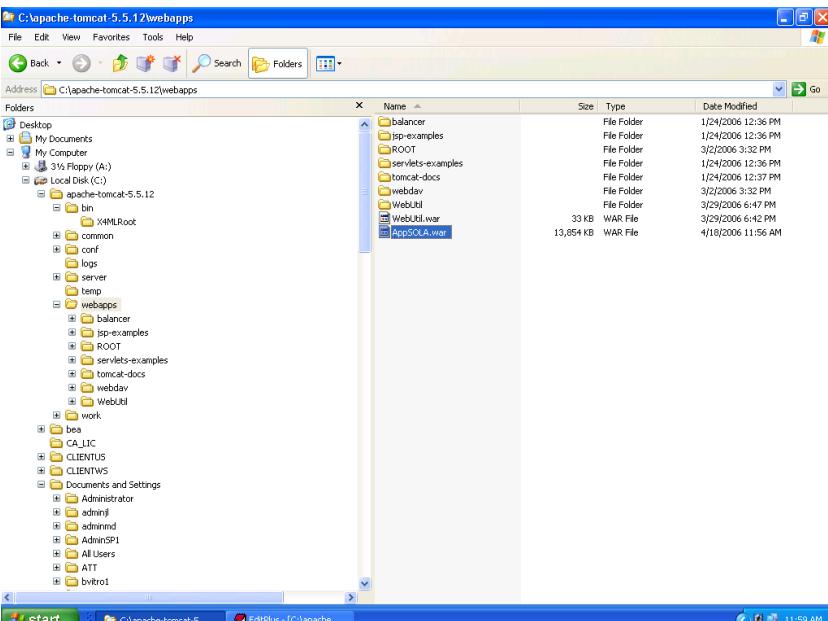
Step	Procedure																														
1.	Insert the SOLA installation package. Change to the /ide/war/ directory.																														
2.	<p>Copy /ide/war/AppSOLA.war from the package to CATALINA_HOME/webapps/AppSOLA.war as shown below: (CATALINA_HOME is where Tomcat is installed)</p>  <table border="1"> <caption>File Explorer Contents</caption> <thead> <tr> <th>Name</th> <th>Type</th> <th>Date Modified</th> </tr> </thead> <tbody> <tr> <td>balancer</td> <td>File Folder</td> <td>1/24/2006 12:36 PM</td> </tr> <tr> <td>jsp-examples</td> <td>File Folder</td> <td>1/24/2006 12:36 PM</td> </tr> <tr> <td>ROOT</td> <td>File Folder</td> <td>3/2/2006 3:32 PM</td> </tr> <tr> <td>servlets-examples</td> <td>File Folder</td> <td>1/24/2006 12:36 PM</td> </tr> <tr> <td>tomcat-docs</td> <td>File Folder</td> <td>1/24/2006 12:37 PM</td> </tr> <tr> <td>webdav</td> <td>File Folder</td> <td>3/2/2006 3:32 PM</td> </tr> <tr> <td>Webddl</td> <td>File Folder</td> <td>3/29/2006 6:47 PM</td> </tr> <tr> <td>Webddl.war</td> <td>WAR File</td> <td>3/29/2006 6:42 PM</td> </tr> <tr> <td>AppSOLA.war</td> <td>WAR File</td> <td>4/16/2006 11:56 AM</td> </tr> </tbody> </table>	Name	Type	Date Modified	balancer	File Folder	1/24/2006 12:36 PM	jsp-examples	File Folder	1/24/2006 12:36 PM	ROOT	File Folder	3/2/2006 3:32 PM	servlets-examples	File Folder	1/24/2006 12:36 PM	tomcat-docs	File Folder	1/24/2006 12:37 PM	webdav	File Folder	3/2/2006 3:32 PM	Webddl	File Folder	3/29/2006 6:47 PM	Webddl.war	WAR File	3/29/2006 6:42 PM	AppSOLA.war	WAR File	4/16/2006 11:56 AM
Name	Type	Date Modified																													
balancer	File Folder	1/24/2006 12:36 PM																													
jsp-examples	File Folder	1/24/2006 12:36 PM																													
ROOT	File Folder	3/2/2006 3:32 PM																													
servlets-examples	File Folder	1/24/2006 12:36 PM																													
tomcat-docs	File Folder	1/24/2006 12:37 PM																													
webdav	File Folder	3/2/2006 3:32 PM																													
Webddl	File Folder	3/29/2006 6:47 PM																													
Webddl.war	WAR File	3/29/2006 6:42 PM																													
AppSOLA.war	WAR File	4/16/2006 11:56 AM																													

Figure 36: Tomcat –war file copy

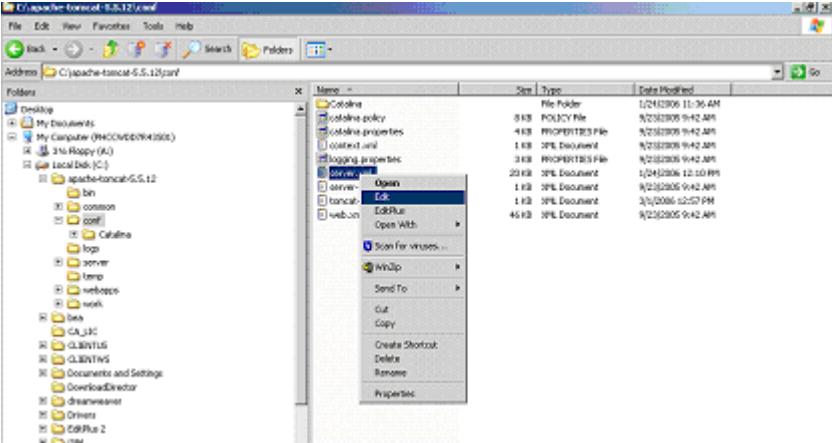
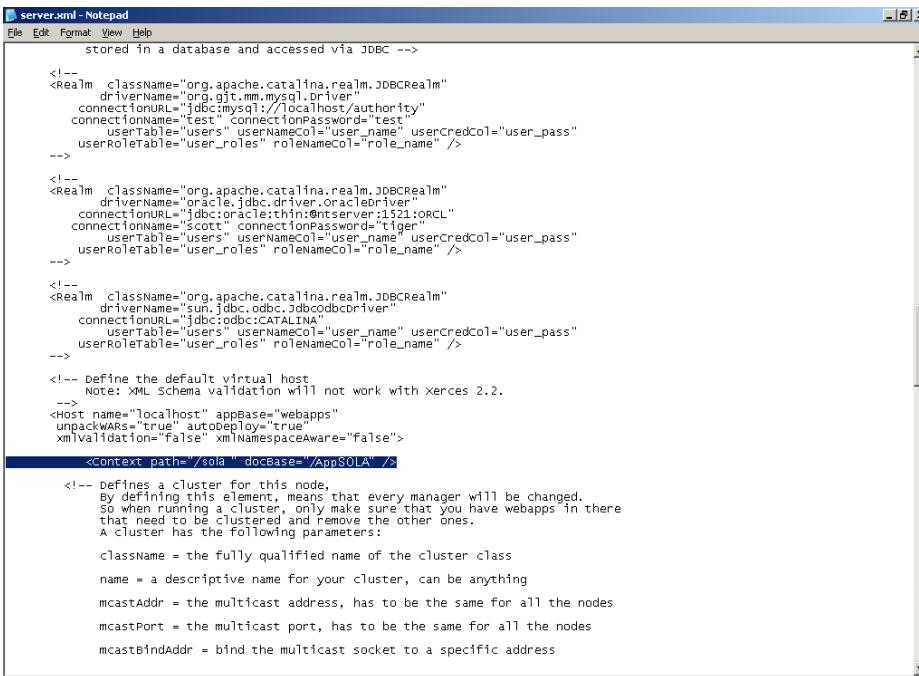
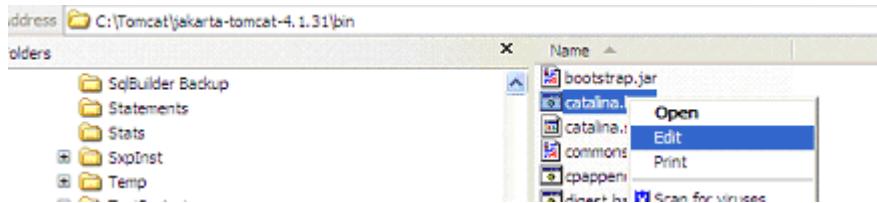
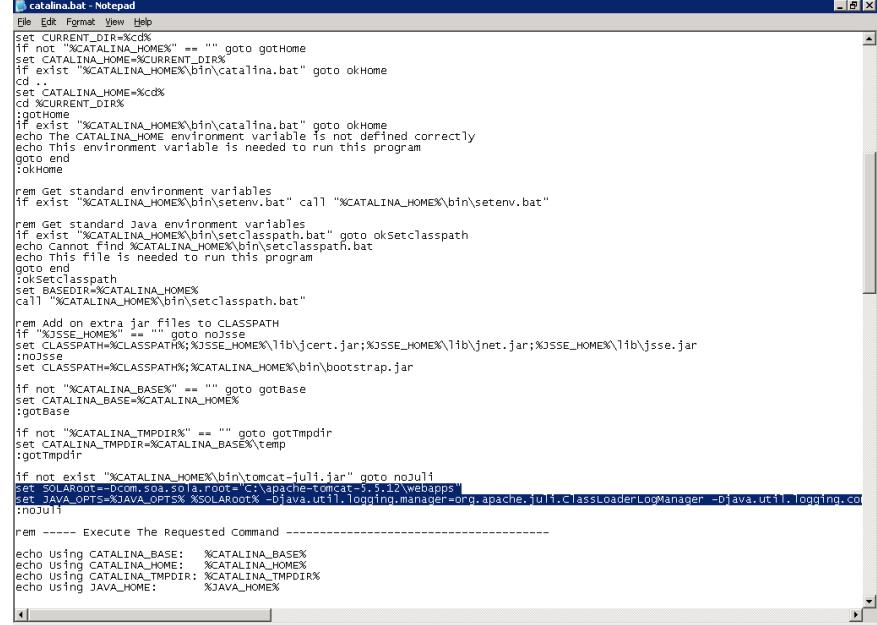
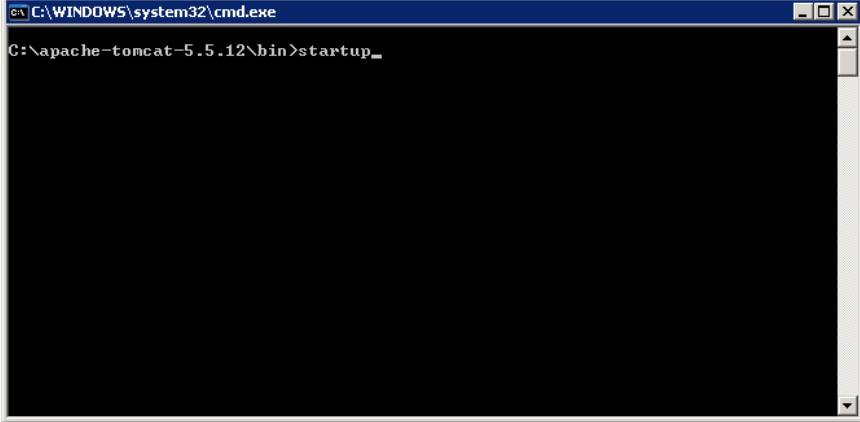
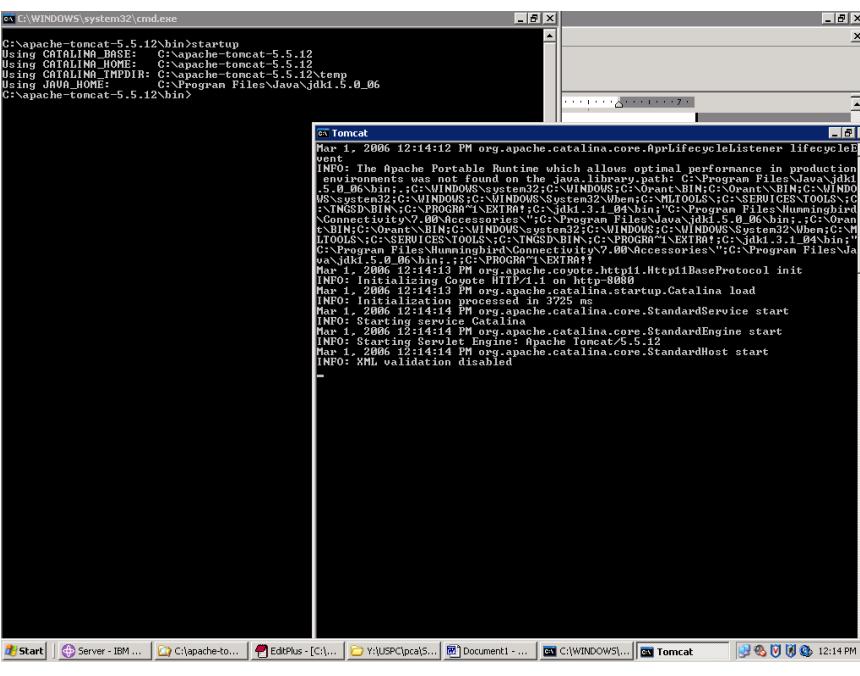
Step	Procedure
3.	<p>Edit the server.xml file (available at CATALINA_HOME\conf) to include a new entry for the Context -- as follows.</p> <p>You may use notepad or any other xml editor to edit server.xml.</p> 
4.	<p>The Root Context should be added inside the <Host> element tag as follows:</p> <pre><Context path="/sola" docBase="/AppSOLA" /></pre> <p>Make sure that Context starts with an uppercase C and docBase has an uppercase B. AppSOLA is the folder under CATALINA_HOME\webapps that contains the code. Save and close the file.</p> 

Figure 37: Tomcat –Edit server.xml

Step	Procedure
5.	<p>Next, edit the catalina.bat file (available at CATALINA_HOME\bin) to add a new variable SOLARoot and modify, or add, the JAVA_OPTS variable as shown below. "SOLARoot" is a property that dictates where SOLA keeps files for its own use.</p>  <p>Figure 39: Tomcat –Edit catalina.bat</p> <p>Add SOLARoot as follows:</p> <pre>set SOLARoot=-Dcom.soa.sola.root=<fileSystem></pre> <p>This will specify where you want SOLA to store product related files. For example, if you already have a directory created called c:\SOLAFiles you would specify SOLARoot as follows:</p> <pre>set SOLARoot= -Dcom.soa.sola.root = "c:\SOLAFiles"</pre> <p>If you do not already have a directory to store SOLA related files then you should first create one.</p> <p>Note: The authority the JVM runs under must have read/write/execute to this file system</p> <p>Next, add the following lines in catalina.bat</p> <pre>set JAVA_OPTS=%JAVA_OPTS% %SOLARoot% - Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager</pre>  <p>Figure 40: Tomcat –catalina.bat</p> <p>Save and close the file.</p>

Step	Procedure
6.	<p>Start the Tomcat Server by invoking the startup script as shown below.</p> 
7.	<p>A new command prompt window should pop up when the server starts.</p> 
8.	<p>Proceed to chapter 4 for customization.</p>

A Client who recently completed installation successfully was quoted:

"We set the sola root to the /SOLARoot directory. We didn't put in the Context path step of the installation because it was causing a startup error. We also didn't rename the war file but instead left it named sola.war which is how it was named upon download from the SOA Support Site."

Chapter 5: Customizing SOLA

This section provides instructions for customizing SOLA for use as an SOA development and run-time platform. The customization steps are performed through your browser using the newly installed Development Studio and the Resource Manager. The Development Studio will in turn customize the SOLA run-time.

Begin by entering values for your installation into the Development Studio Customization worksheet on the next page.

DEVELOPMENT STUDIO CUSTOMIZATION WORKSHEET

Parameter Name in the sample WRKSHEET provided	Description	Notes/Examples	Your Values
SOLA Website	This is the http address where SOLA Web-site is accessible	http://sola.web.com:9203/sola/index.html The index page is available under context sola. Hence, append /sola/index.jsp to the server FQDN.	
productKey	Supplied product key	A3zUx8iLqprTm0	
FTPSite	Mainframe FTP FQDN	DEVMF.SYSPLEX.COM	
SOLASoapAddress	The FQDN of a SOLA Container which provides the mainframe backend for the SOLA Development Studio. If the container is a SOLA CICS Container then append "/CICS/XML/XMLPC000" at the end of FQDN where CICS is listening. If the container is SOLA IMS Container then append "/XMLPC000" at the end of FQDN where SOLA is listening. This region may or may not be the same as the OpenAccessEndPoint FQDN.	 http://cicsFQDN:3067/CICS/XML/XMLPC000 for a SOLA CICS Container or http://SOLAfQDN:3067/XMLPC000 for a SOLA IMS Container	
MQTestEndPoint (repeat for each endpoint)	The queue name that SOLA will use for receipt of SOAP over MQ messages. This value is optional.		
OpenAccessEndPoint (repeat for each SOLA Container)	The FQDN of a SOLA Container. This region is will be accessible to anyone from the web-site. It may or may not be the same as the SOLASoapAddress FQDN. If the container is a SOLA CICS Container then append "/CICS/XML/XMLPC000" at the end of FQDN where CICS is listening. If the container is SOLA IMS Container then append "/XMLPC000" at the end of FQDN where	 http://cicsFQDN:3067/CICS/XML/XMLPC000 for a SOLA CICS Container or http://SOLAfQDN:3067/XMLPC000 for a SOLA IMS Container	

	SOLA is listening.		
RestrictedAccessEndPoint (repeat for each SOLA Container)	The FQDN of a SOLA Container. This region will only be accessible to authorized users of the Development Studio. If the container is a SOLA CICS Container then append "/CICS/XML/XMLPC000" at the end of FQDN where CICS is listening. If the container is SOLA IMS Container then append "/XMLPC000" at the end of FQDN where SOLA is listening.. .	 http://cicsFODN:3067/CICS/XML/XMLPC000 for a SOLA CICS Container or http://SOLAfODN:3067/XMLPC000 for a SOLA IMS Container	
IP Address/FQDN:	The hostname or IP Address of the TCPIP stack that IMS Connect connects to.	10.20.12.5	
Port	The port that IMS Connect is listening on.	1255	
Data Store Id:	The Datastore name corresponding to the configuration parameter "ID" in the "DATASTORE" configuration statement of IMS Connect.	IMSCON	
TCP/IP Stack Name	The name of the TCPIP stack that IMS Connect connects to.	MVSTCP	
Number of Sessions	Max number of connections enabled with IMS Connect. This parameter is ignored with this release.	1	
IMS Group Name	IMS XCF Group name as defined by parm GRNAME in the IMS subsystem.	IMSTST1	
OTMA Name	The name that IMS will have within the XCF group (specified by parm OTMNAM in the IMS subsystem).	IMSTST1	
OTMA Client Name	Unique user defined name with which the OTMA client will be defined in the OTMA connection (8 characters)	IMSOTMA	
OTMA TPipe Prefix	4 character prefix for the Transaction Pipe enabled for the OTMA sessions.	IMS1	

Num of Sessions	Max number of concurrent sessions enabled with the OTMA connection.	4	
-----------------	---	---	--

Next follow the steps below to customize the SOLA installation and make it ready for use as an SOA development and run-time platform.

Note 1: The z/Series mainframe installation must be completed before you can customize the SOLA Development Studio.

Note 2: If you are an existing SOLA customer and are upgrading from SOLA 5.1 (or if you are a SOLA 6.0 customer), then the migration process will have copied your SOLA 5.1 configuration information and you won't need to follow the customization steps in this chapter. The migration process is documented in the SOLA 6.1 Migration Guide.

Note 3: If you are a SOLA 6.0 customer, you should restart the SOLA Server at this point to complete your installation steps.

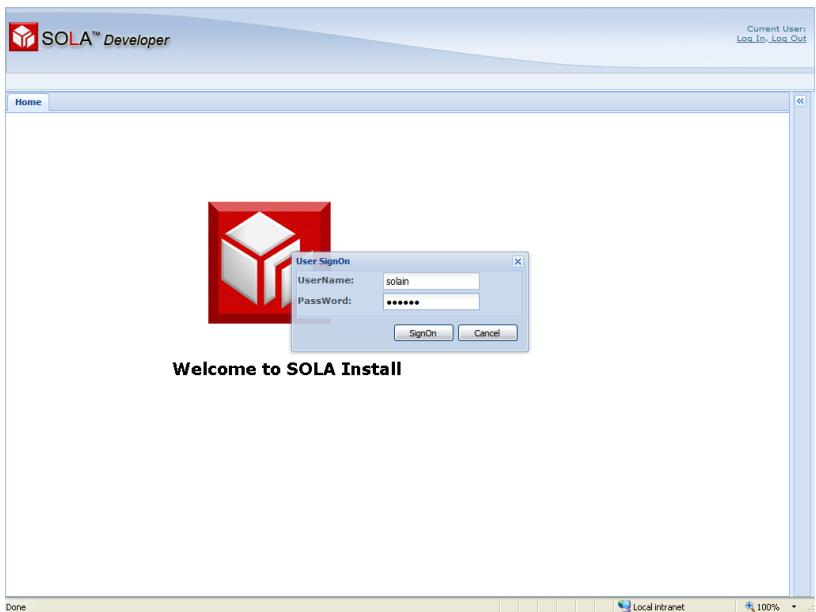
Step	Procedure
1.	Enter http://servername:port/sola/install.html on your browser where <i>servername</i> and <i>port</i> refer to the server (WebSphere, WebLogic or Tomcat) where you installed the SOLA Development Studio.
2.	<p>This will display the following page. Enter <u>solain</u> as the UserName and <u>solain</u> as the password.</p> 

Figure 43: Login with the installation ID

3. You will be prompted for the product key. Please type the product key on the prompt as shown below. The product key will be provided separately.

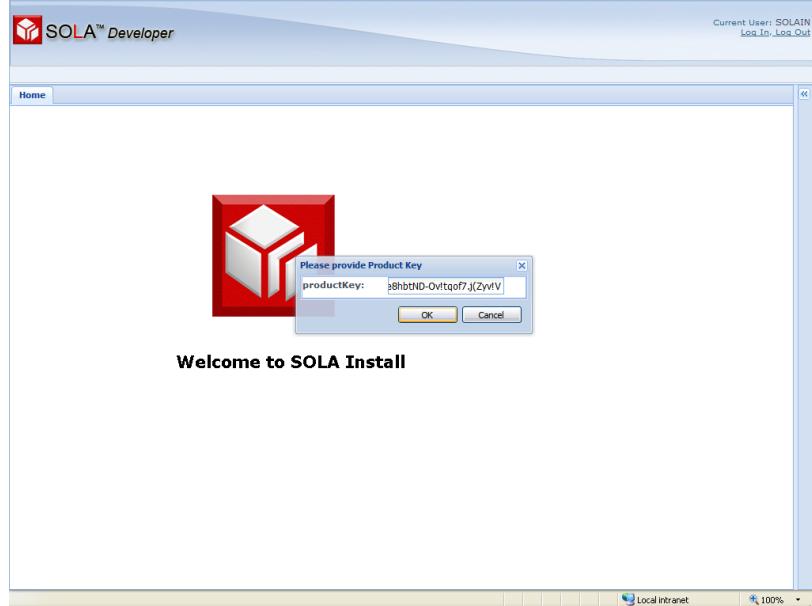


Figure 44: Enter the product ID

4. Read and agree to the license agreement.



Figure 45: Read and agree to the license agreement

5. Accept the license agreement by SCROLLING TO THE BOTTOM and selecting "I accept the terms of the license agreement" and click "Submit."

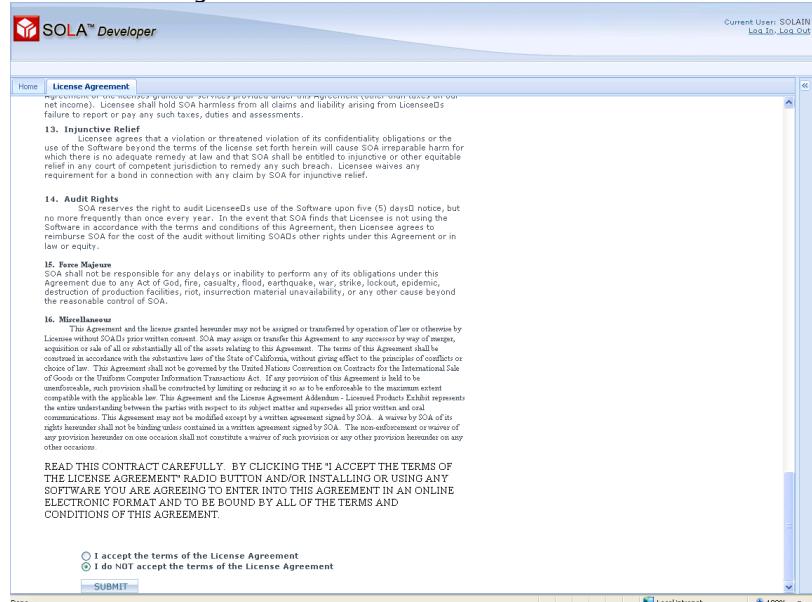


Figure 46: Accept the license agreement

6. The customization panel will be displayed, beginning with the debugging.xml file.

If the debugging.xml properties are not automatically displayed, go to the Property Editor and choose **/inst /system /debugging.xml** from the three dropdown boxes.

Properties	Allowable Values
installationPassword	Encrypted value is pre-filled here. Do not modify on this page, changing the installation password is done on a separate page.
InstallationUserId*	Optional. Enter a valid SAF ID. SOLA will use this ID to run all back end transactions that support the Development Studio. * This optional property should only be defined if you have completed the optional step described in section Define the CICS Analyzer to use with SOLA (Optional) on page 34.
consoleFile	YES – creates and append stderror.txt and stdout.txt with error or warning messages. NO – sends messages to the server console.
Debug	I - Log all informative, warnings & Errors W - Log all warnings and error messages E - Log only error Messages. Value "E" is default and recommended.
productKey	SOLA's product key. This will be prefilled based on the key you entered previously.

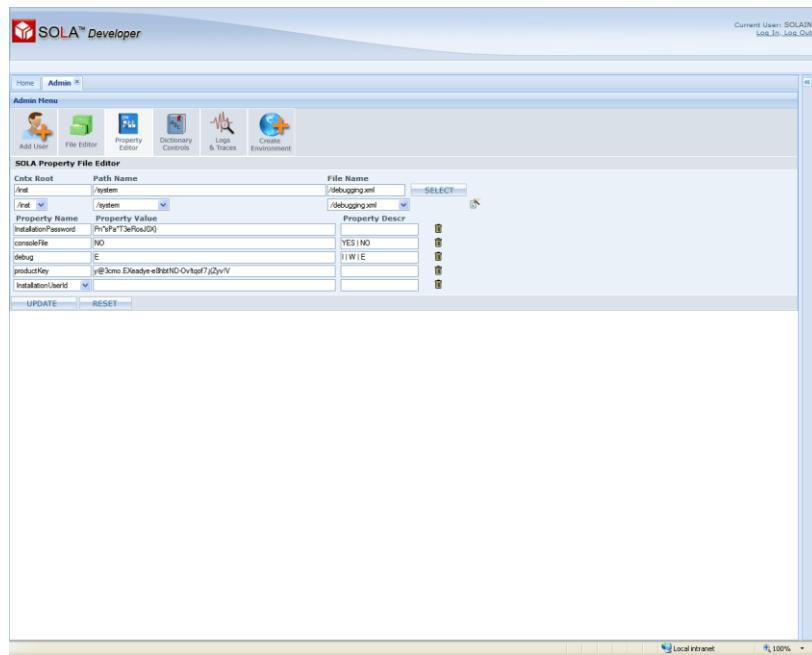
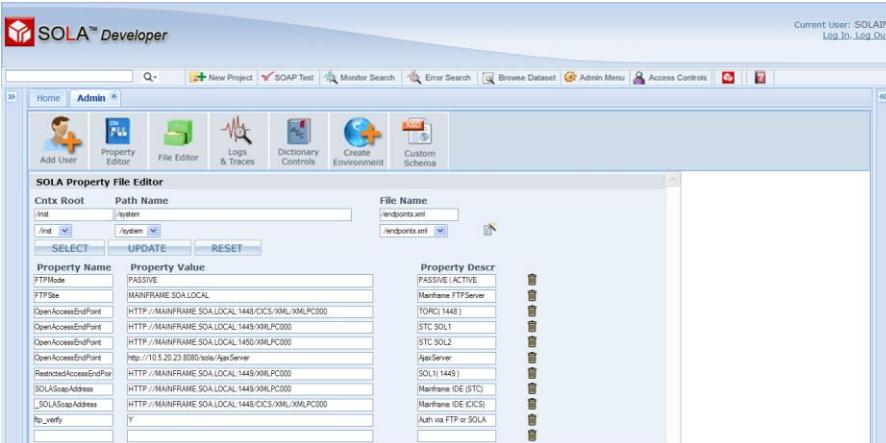


Figure 47: Debugging XML

7.	<p>Now you will need to set up the endpoints that the SOLA run-time will manage. As shown in the screen shot below, click on the drop down box below the "Filename" heading and choose "/endpoints.xml".</p>  <table border="1" data-bbox="470 783 1356 819"> <thead> <tr> <th>Property Name</th><th>Property Value</th><th>Property Descr</th></tr> </thead> <tbody> <tr> <td>FTPMode</td><td>PASSIVE</td><td>Mainframe FTPServer</td></tr> <tr> <td>FTPSite</td><td>MAINFRAME SOA LOCAL</td><td>TODI(1448)</td></tr> <tr> <td>OpenAccessEndPoint</td><td>HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000</td><td>STC SOL1</td></tr> <tr> <td>OpenAccessEndPoint</td><td>HTTP://MAINFRAME.SOAL.LOCAL:1449/XML/PC000</td><td>STC SOL2</td></tr> <tr> <td>OpenAccessEndPoint</td><td>HTTP://MAINFRAME.SOAL.LOCAL:1450/XML/PC000</td><td>AppServer</td></tr> <tr> <td>OpenAccessEndPoint</td><td>http://10.5.20.23:8080/webs/AxisServer</td><td>SOL1(1449)</td></tr> <tr> <td>RestrictedAccessEndPoint</td><td>HTTP://MAINFRAME.SOAL.LOCAL:1449/XML/PC000</td><td>Mainframe IDE (STC)</td></tr> <tr> <td>SOLAsoapAddress</td><td>HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000</td><td>Mainframe IDE (CICS)</td></tr> <tr> <td>SOLASnapAddress</td><td>HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000</td><td>Auth via FTP or SOLA</td></tr> <tr> <td>Rp_verify</td><td>Y</td><td></td></tr> </tbody> </table>	Property Name	Property Value	Property Descr	FTPMode	PASSIVE	Mainframe FTPServer	FTPSite	MAINFRAME SOA LOCAL	TODI(1448)	OpenAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000	STC SOL1	OpenAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1449/XML/PC000	STC SOL2	OpenAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1450/XML/PC000	AppServer	OpenAccessEndPoint	http://10.5.20.23:8080/webs/AxisServer	SOL1(1449)	RestrictedAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1449/XML/PC000	Mainframe IDE (STC)	SOLAsoapAddress	HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000	Mainframe IDE (CICS)	SOLASnapAddress	HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000	Auth via FTP or SOLA	Rp_verify	Y	
Property Name	Property Value	Property Descr																																
FTPMode	PASSIVE	Mainframe FTPServer																																
FTPSite	MAINFRAME SOA LOCAL	TODI(1448)																																
OpenAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000	STC SOL1																																
OpenAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1449/XML/PC000	STC SOL2																																
OpenAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1450/XML/PC000	AppServer																																
OpenAccessEndPoint	http://10.5.20.23:8080/webs/AxisServer	SOL1(1449)																																
RestrictedAccessEndPoint	HTTP://MAINFRAME.SOAL.LOCAL:1449/XML/PC000	Mainframe IDE (STC)																																
SOLAsoapAddress	HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000	Mainframe IDE (CICS)																																
SOLASnapAddress	HTTP://MAINFRAME.SOAL.LOCAL:1448/CICS/XML/PC000	Auth via FTP or SOLA																																
Rp_verify	Y																																	
8.	<p>The key-value pairs will change to reflect the values in the endpoints.xml file. In this screen you will need to specify an FQDN for the following key-value pairs in the <i>endpoints.xml</i> file.</p> <table border="1" data-bbox="372 952 1455 1860"> <thead> <tr> <th>Key</th><th>Value</th></tr> </thead> <tbody> <tr> <td>FTPMode</td><td>ACTIVE or PASSIVE. Determines which FTP mode the SOLA Development Studio will use for communication with the mainframe FTP server for retrieving and storing datasets. ACTIVE recommended.</td></tr> <tr> <td>FTPSite</td><td>The SOLA Development Studio communicates with z/OS in two ways – via SOLA web services for all directory access and via FTP for access to z/OS datasets, JES output and the JES scheduler. In this value you need to specify the address of the z/OS FTP server.</td></tr> <tr> <td>SOLAsoapAddress</td><td>The FQDN of a SOLA Container. This region provides the mainframe backend for the SOLA Development Studio. This region may or may not be the same as the OpenAccessEndPoint FQDN.</td></tr> <tr> <td>OpenAccessEndPoint</td><td>The FQDN of a SOLA Container. This region is where developers test their services. This region may or may not be the same as the <u>SOLAsoapAddress</u> FQDN.</td></tr> <tr> <td>MQTestEndPoints</td><td>Optional. The queue name that SOLA will use for receipt of SOAP over MQ messages. This value is optional.</td></tr> <tr> <td>RestrictedAccessEndPoint</td><td>Optional. The FQDN of a secured SOLA Container. This region is where services are executed. Only administrators will be able to see this value in an endpoint dropdown. This value is optional.</td></tr> </tbody> </table>	Key	Value	FTPMode	ACTIVE or PASSIVE. Determines which FTP mode the SOLA Development Studio will use for communication with the mainframe FTP server for retrieving and storing datasets. ACTIVE recommended.	FTPSite	The SOLA Development Studio communicates with z/OS in two ways – via SOLA web services for all directory access and via FTP for access to z/OS datasets, JES output and the JES scheduler. In this value you need to specify the address of the z/OS FTP server.	SOLAsoapAddress	The FQDN of a SOLA Container. This region provides the mainframe backend for the SOLA Development Studio. This region may or may not be the same as the OpenAccessEndPoint FQDN.	OpenAccessEndPoint	The FQDN of a SOLA Container. This region is where developers test their services. This region may or may not be the same as the <u>SOLAsoapAddress</u> FQDN.	MQTestEndPoints	Optional. The queue name that SOLA will use for receipt of SOAP over MQ messages. This value is optional.	RestrictedAccessEndPoint	Optional. The FQDN of a secured SOLA Container. This region is where services are executed. Only administrators will be able to see this value in an endpoint dropdown. This value is optional.																			
Key	Value																																	
FTPMode	ACTIVE or PASSIVE. Determines which FTP mode the SOLA Development Studio will use for communication with the mainframe FTP server for retrieving and storing datasets. ACTIVE recommended.																																	
FTPSite	The SOLA Development Studio communicates with z/OS in two ways – via SOLA web services for all directory access and via FTP for access to z/OS datasets, JES output and the JES scheduler. In this value you need to specify the address of the z/OS FTP server.																																	
SOLAsoapAddress	The FQDN of a SOLA Container. This region provides the mainframe backend for the SOLA Development Studio. This region may or may not be the same as the OpenAccessEndPoint FQDN.																																	
OpenAccessEndPoint	The FQDN of a SOLA Container. This region is where developers test their services. This region may or may not be the same as the <u>SOLAsoapAddress</u> FQDN.																																	
MQTestEndPoints	Optional. The queue name that SOLA will use for receipt of SOAP over MQ messages. This value is optional.																																	
RestrictedAccessEndPoint	Optional. The FQDN of a secured SOLA Container. This region is where services are executed. Only administrators will be able to see this value in an endpoint dropdown. This value is optional.																																	

	dataSource	<p>This property allows customers to control where the SOLA Developer and Resource Manager store and retrieve internal meta data.</p> <p>There are two possible values:</p> <ol style="list-style-type: none"> 1. SOLA – (the default if this property is not specified). Web services calls to SOLA's mainframe execution environment are made to store and retrieve data from a DB2 Database 2. FS – This option will use the local file system to store and retrieve the meta data.
	ftp_verify	<p>Optional. The default value of this property is 'N'.</p> <p>This property allows customers to control how user logons to the IDE are verified against RACF or equivalent.</p> <p>By default the SOLA mainframe runtime processes the logon request.</p> <p>For SOLA CICS Container leave the default as 'N'.</p> <p>For SOLA IMS Container (Started Task) adjust the value as follows</p> <ul style="list-style-type: none"> • If the started task is configured to run in Authorized state [Refer to "<i>Chapter 3: Customizing SOLA IMS Container on a z/Series mainframe</i>", Section: APF AUTHORIZES THE SOLA LOAD LIBRARY (Recommended)] then leave the property to default 'N'. • If the started task is configured to run in Unauthorized state then set the value to 'Y' so the FTP mechanism will be used to verify user credentials at logon
	UDDIServerAddress	Optional. The FQDN of a central UDDI V3 registry for publishing SOLA Services. Release 6.1 of SOLA supports Service Manager from SOA Software.
	UDDIUser	Optional. The UserId to use to connect to the UDDIServer.
	UDDIPassword	Optional. The password associated with the UDDI UserId.

9. In the screen below we've entered a value for *FTPMode*, *FTPSite*, *SOLASoapAddress*, *OpenAccessEndPoint* and *dataSource*. We gave the *SOLASoapAddress* a "Property Descr" of *SOLA Development Studio* and *OpenAccessEndPoint* a "Property Descr" of *TEST*. These values will be used in the endpoint URL drop down.

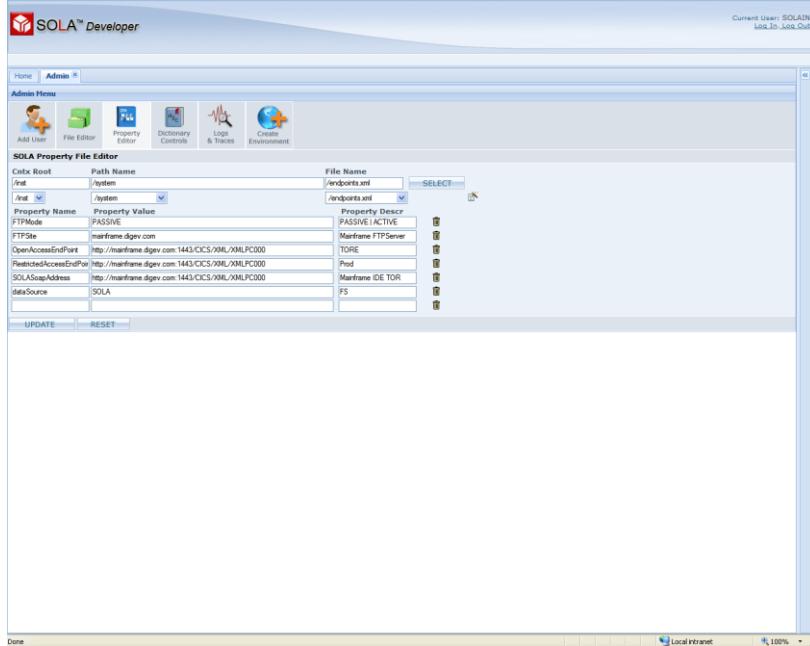
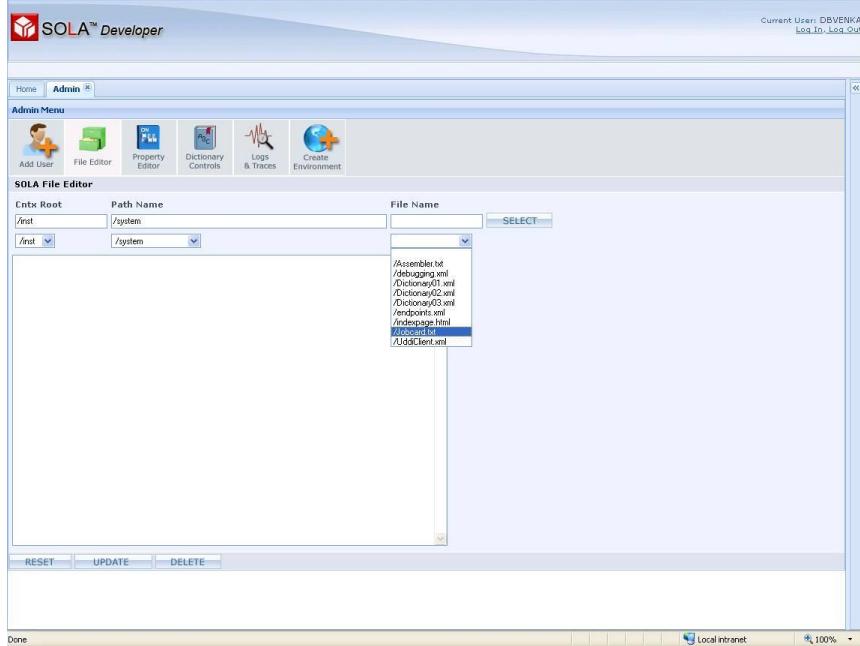


Figure 49: endpoints changed

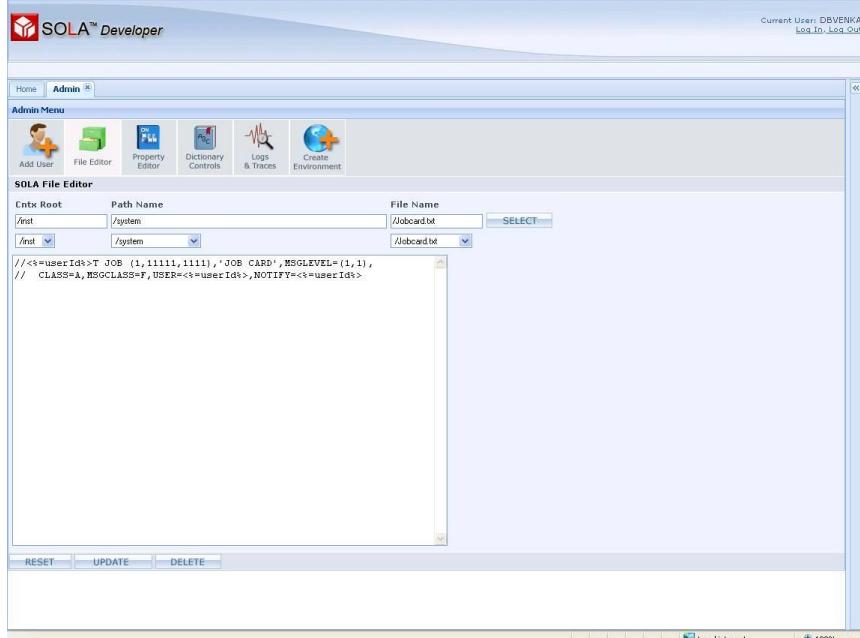
In this example *RestrictedEndPoint*, *UDDIServerAddress*, *UDDIUser* and *UDDIPassword* properties have not been filled. You must delete these properties by clicking trash bin icon on the right if you don't intend to use these values. You can add them back at any time.

Click the **Update** button to update the file and propagate the changes.

10. Next you'll need to verify, and optionally modify the JCL Jobcard that SOLA will use to submit jobs to assemble Templates (SOLA's runtime metadata). Go to the File Editor and choose “/inst” from the Cntx Root dropdown, “/system” from the Path Name dropdown and “/Jobcard.txt” from the File Name dropdown.

**Figure 50: Selecting /Jobcard.txt**

11. Modify the accounting, CLASS and MSGCLASS parameters to meet your installation's requirements.

**Figure 51: Modifying /Jobcard.txt**

Click the **Update** button to update the file and propagate the changes.

12. Next you'll need to create environments that conform to your Software Development Life Cycle. We recommend you create at least three environments "T" (test), "S" (stage) and "P" (production) with sequences 1, 10 and 20 respectively.

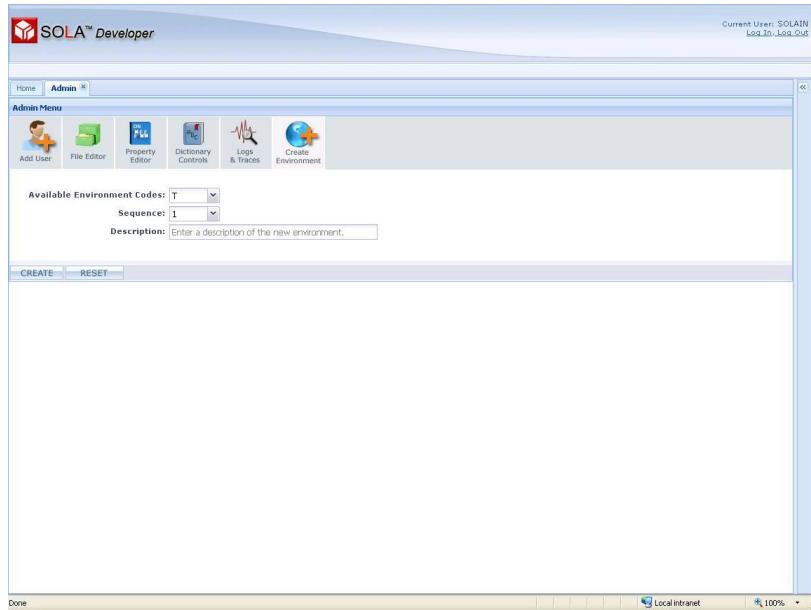


Figure 52: Creating Environments

13. The next step is to add an administrator. Click on the Add User icon and enter the RACF Id for the SOLA Administrator, then click the Create button.

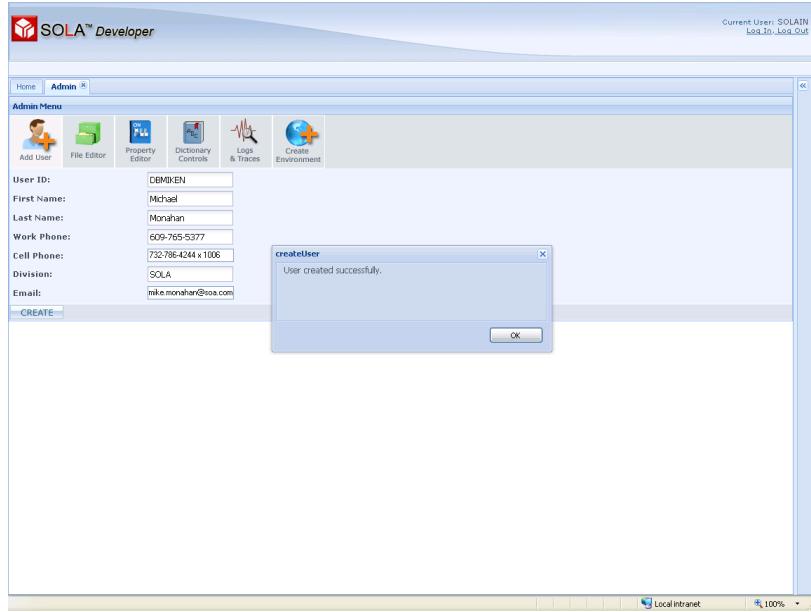


Figure 53: Adding the SOLA Administrator

14. The next step is to configure each of the endpoints that we entered into the endpoints.xml file. You use the SOLA Resource Manager to configure endpoints. Enter the following url: <http://<yourservername>:<port>/sola/ResourceMgr.html>.

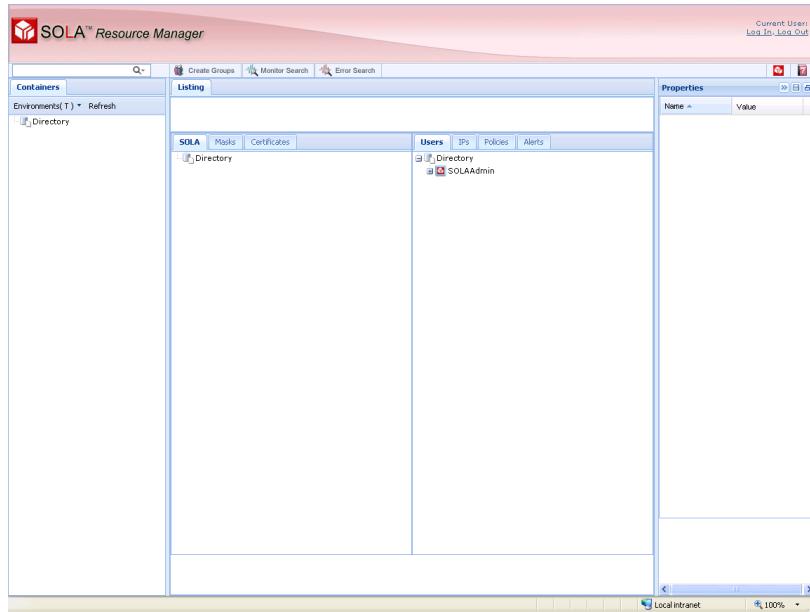


Figure 54: Creating a Container group and Container

15. Pick the environment you want to create Container group for. Let's assume you want to create a new container group in environment "T" which is already listed on the top and selected so you don't need to do anything special.

Now right mouse click on the container directory and pick create tor group menu

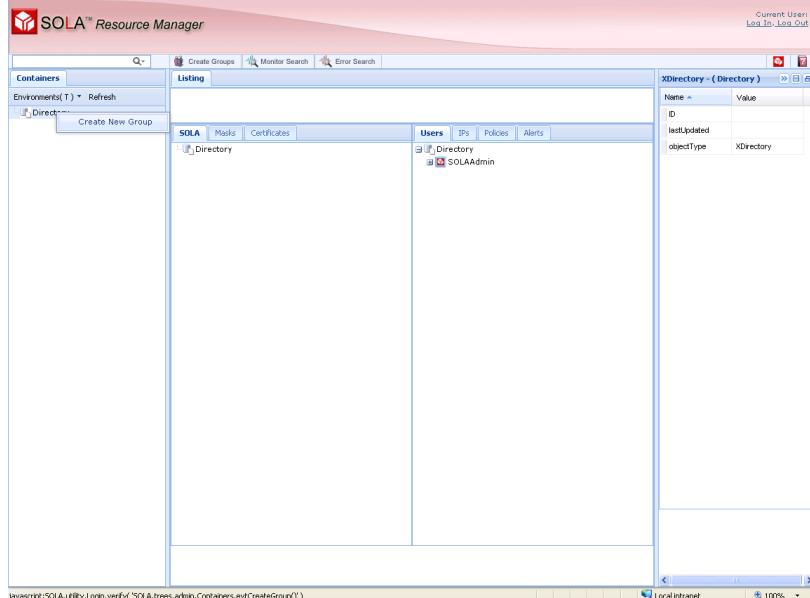


Figure 55: Choosing the environment

16. You'll be prompted to login with a RACF Id. Use the SOLA Administrator Id that you just created.

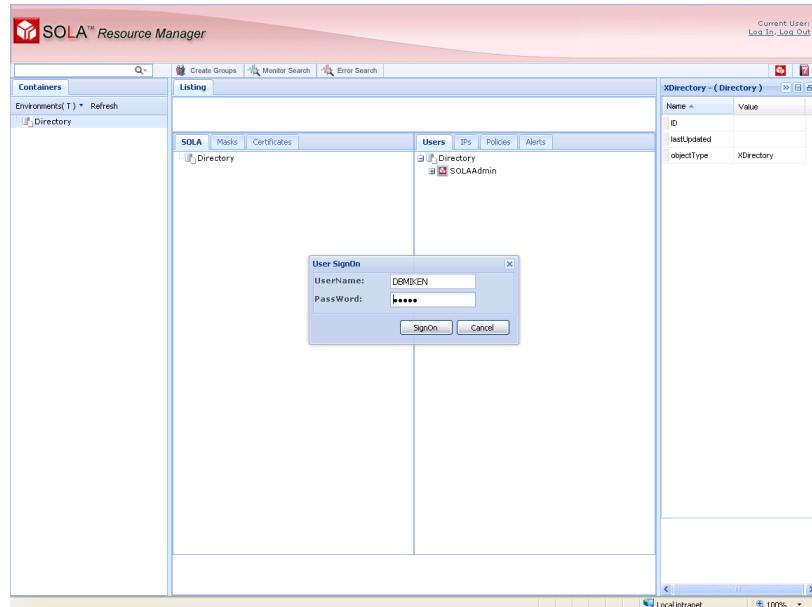


Figure 56: Login with the SOLA Administrator Id

17. The “Create Container group” page will be displayed.

Figure 57: Create Container Group

We called our new group ‘SOLA Test1 Regions’.

The blue background part of the screen contains the settings for the group of containers. You can change these settings, but for the moment we recommend that you leave them at the default settings.

The bottom half of the screen (white background) is split into two parts. The upper part contains the default policy settings for the group. We recommend that you leave them at the default settings. The lower part contains the IMS connection information, and this is used to define how the SOLA container is to connect to IMS for the optional IMS component. Defining the IMS connection information is covered below.

Press the create button to create the Container Group.

18. If your SOLA package includes the optional IMS component then you will need to complete the lower part of the screen to choose whether SOLA should use IMS Connect or OTMA/CI to connect to IMS.

Note: If your IMS transactions need to run under a RACF ID then you will need to choose IMS Connect. It isn’t possible to pass credentials from CICS to IMS when using OTMA/CI.

Figure 58: Create Container Group (IMS version)

Choose how you want to connect to IMS from the dropdown.

You should complete *either* the IMS Connect information or the OTMA information, but not both.

The fields for IMS Connect are:

- IP Address/FQDN:** The hostname/IP Address of the TCPIP stack that IMS Connect connects to.
- Port:** The port that IMS Connect is listening on.
- Data Store Id:** The Datastore name corresponding to the configuration parameter "ID" in the "DATASTORE" configuration statement of IMS Connect.
- TCP/IP Stack Name:** The name of the TCPIP stack that IMS Connect connects to.
- Number of Sessions:** Max number of connections enabled with IMS Connect. This parameter is ignored with this release.

The fields for OTMA are:

- IMS Group Name:** IMS XCF Group name as defined by parm GRNAME in the IMS subsystem.
- OTMA Name:** Specified the name that IMS will have within the XCF group (specified by parm OTMNAM in the IMS subsystem).
- OTMA Client Name:** Unique user defined name with which the OTMA client will be defined in the OTMA connection (8 characters)
- OTMA TPipe Prefix:** 4 character prefix for the Transaction Pipe enabled for the OTMA sessions.
- Num of Sessions:** Max number of concurrent sessions enabled with the OTMA connection.

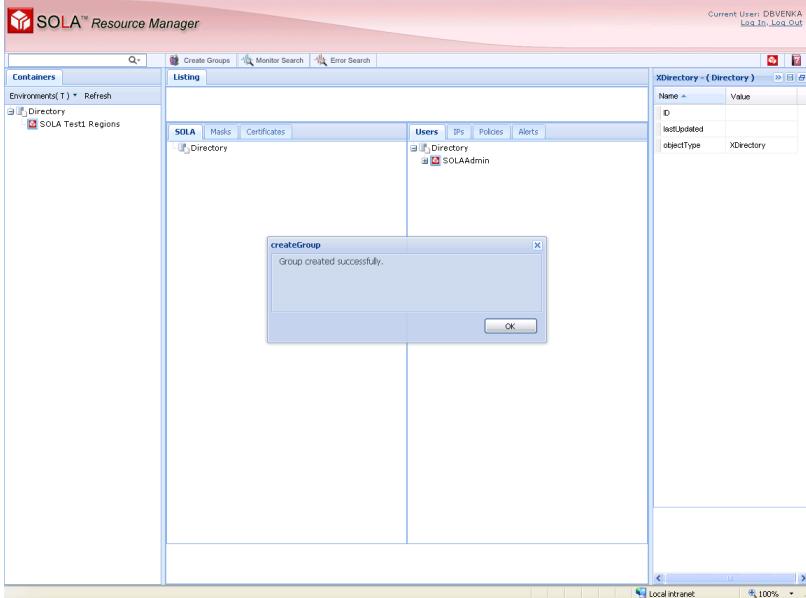
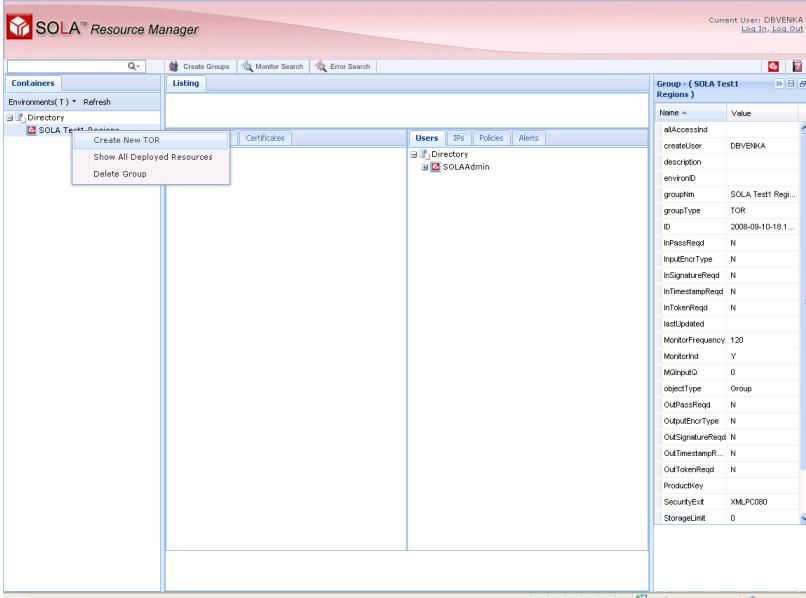
	<p>Press the create button to create the Container Group.</p>
19.	<p>The new group 'SOLA Test1 Regions' has been created, and is shown in the directory tree in the directory pane.</p>  <p>The screenshot shows the SOLA Resource Manager interface. In the left sidebar under 'Containers', there is a 'Directories' section with a node labeled 'SOLA Test1 Regions'. The main pane is titled 'Listing' and contains tabs for 'SOLA', 'Masks', 'Certificates', 'Users', 'IPs', 'Policies', and 'Alerts'. Under 'Users', there is a single entry for 'SOLAAdmin'. A modal dialog box titled 'createGroup' is displayed in the center, stating 'Group created successfully.' with an 'OK' button. On the right side of the screen, there is a 'XDirectory - (Directory)' pane showing a table with columns 'Name' and 'Value'. The table has one row with ID 'SOLA Test1 Regions' and object type 'XDirectory'. The status bar at the bottom indicates 'Local intranet' and '100%'.</p>
20.	<p>The next step is to create a SOLA Container as a member of the group. Right click on the new group and select "Create New TOR". Before creating a new TOR (container), make sure you have the URL for the SOLA region and it is ready to accept requests on the mainframe.</p>  <p>The screenshot shows the same SOLA Resource Manager interface as above. The 'SOLA Test1 Regions' group is selected, and a context menu is open over it. The menu options are 'Create New TOR', 'Show All Deployed Resources', and 'Delete Group'. The main pane and right-hand XDirectory pane are visible in the background.</p>

Figure 59: Container Group created successfully

Figure 60: Create a new container

21. Fill in the region specific details.

Name	Value
allAccessId	DBVENKA
creatdUser	DBVENKA
description	
enviroID	
groupNm	SOLA Test1 Regi...
groupType	TOR
ID	2008-09-10-18.1...
InPassReqd	N
InPduIncrType	N
InSignatureReqd	N
InTimestampReqd	N
InTokenReqd	N
IsL1Updated	
MonitorFrequency	120
MonitorId	Y
MoObjectQ	0
objectType	Group
OutPassReqd	N
OutputEnvrType	N
OutSignatureReqd	N
OutTimestampR...	N
OutTokenReqd	N
ProductKey	
SecureExt	XMLPC080
StorageUnit	0

Figure 61: Create a new container

The following is a description of the fields:

Sysid: the 4 character system Id of a region.

TOR System Name: the name of a SOLA region.

EndPoint: the soap end point of the SOLA region. To obtain this information, ask your system administrator for the URL that is required to access this region. The address uses the following format:

SOLA CICS Container:

http://mainframe IP address or FQDN:portnumber/CICS/XML/XMLPC000

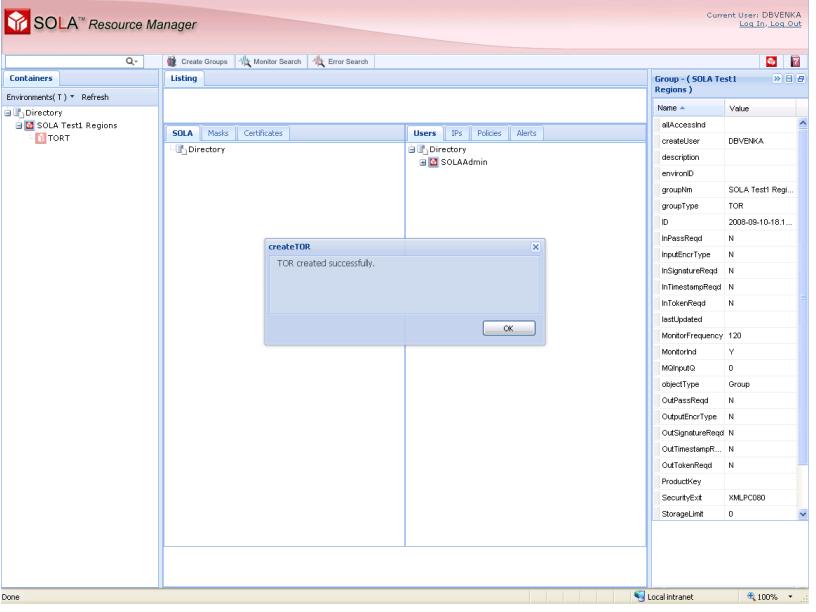
SOLA IMS Container:

http://mainframe IP address or FQDN:portnumber/ XMLPC000

When you have provided all of the necessary information, click **insert** to create the listener group and region. You will be prompted with a confirmation dialog box.

Description: a free-form description of the SOLA region.

When you have provided all of the necessary information, click **Create** to create the TOR.

22.	<p>SOLA will attempt to create a SOLA runtime container in the TOR. If SOLA is able to connect to the region and create the container then a confirmation message will be displayed.</p> 
23.	<p>SOLA is now ready for use. Please close your browser window and open a new instance, logging in with the Admin Id you just created. Test your installation by executing the samples shipped with the product, you can find information on executing the samples by going through the tutorials in the SOLA User Guide.</p>