# Artix™ Mainframe

## Installation Guide

Version 5.1, December 2007

Making Software Work Together™

IONA Technologies PLC and/or its subsidiaries may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this publication. Except as expressly provided in any written license agreement from IONA Technologies PLC, the furnishing of this publication does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Any rights not expressly granted herein are reserved.

IONA, IONA Technologies, the IONA logos, Orbix, Artix, Making Software Work Together, Adaptive Runtime Technology, Orbacus, IONA University, and IONA XMLBus are trademarks or registered trademarks of IONA Technologies PLC and/or its subsidiaries.

Java and J2EE are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. CORBA is a trademark or registered trademark of the Object Management Group, Inc. in the United States and other countries. All other trademarks that appear herein are the property of their respective owners.

IONA Technologies PLC makes no warranty of any kind to this material including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. IONA Technologies PLC shall not be liable for errors contained herein, or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

COPYRIGHT NOTICE

No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording or otherwise, without prior written consent of IONA Technologies PLC. No third party intellectual property right liability is assumed with respect to the use of the information contained herein. IONA Technologies PLC assumes no responsibility for errors or omissions contained in this book. This publication and features described herein are subject to change without notice.

Copyright © 2001-2007 IONA Technologies PLC. All rights reserved.

All products or services mentioned in this manual are covered by the trademarks, service marks, or product names as designated by the companies who market those products.

Updated: December 13, 2007

# Contents

# Preface

## What is Covered in This Book

This book describes the prerequisites and procedures for installing Artix Mainframe on z/OS.

## Who Should Read This Book

This book is intended for z/OS systems programmers who want to install the mainframe components of Artix Mainframe.

The off-host components of Artix Mainframe are installed with Artix for Windows, Linux, and Solaris platforms. See the Artix ESB *Installation Guide* for information on installing these components.

## Prerequisites

The supported platforms, supported compilers, and other prerequisites to using Artix Mainframe are described in "Installation Prerequisites" on page 15.

## The Artix Mainframe Library

The Artix Mainframe documentation library includes the following books:

- Introduction to Artix Mainframe
- Getting Started
- Service-Enabling Existing z/OS Applications
- Creating New z/OS Applications from WSDL
- Creating New z/OS Applications from IDL
- Service-Enabling DB2 SQL
- Common User Tasks
- Administrator's Guide

### Introduction to Artix Mainframe

The *Introduction to Artix Mainframe* guide provides an introductory overview of Artix Mainframe in terms of its components and the various integration solutions that it supports. It also provides an introductory overview of Web services and CORBA technology in general.

### Getting Started

The *Getting Started* guide is intended to help you get started quickly with Artix Mainframe. It provides demonstration walkthroughs of the various integration solutions that Artix Mainframe supports.

### Service-Enabling Existing z/OS Applications

The *Service-Enabling Existing z/OS Applications* guide explains how to use Artix Mainframe to expose existing mainframe applications as Web services or CORBA servers or both.

### Creating New z/OS Applications from WSDL

The *Creating New z/OS Applications from WSDL* guide explains how to use Artix Mainframe to create CICS or IMS-based COBOL or PL/I Web services starting from WSDL. It also explains how to create z/OS-based COBOL or PL/I Web service clients starting from WSDL.

### Creating New z/OS Applications from IDL

The *Creating New z/OS Applications from IDL* guide explains how to use Artix Mainframe to create z/OS-based COBOL or PL/I Artix CORBA clients starting from IDL.

### Service-Enabling DB2 SQL

The *Service-Enabling DB2 SQL* guide explains how to use Artix Mainframe to expose DB2 SQL statements and stored procedures as Web services.

### Common User Tasks

The *Common User Tasks* guide explains how to use Artix Mainframe to perform various common tasks and implement various features that are not specific to any particular integration solution. It is provided as an addendum to the other user guides.

### Administrator's Guide

The *Administrator's Guide* describes how to set up and configure the various features of Artix Mainframe. It also discusses the various levels of security that can be used in the product.

## Getting the Latest Version

The latest updates to the Artix Mainframe documentation can be found at http://www.iona.com/support/docs.

Compare the version dates on the web page for your product version with the date printed on the copyright page of the PDF edition of the book you are reading.

## Searching the Artix Library

You can search the online documentation by using the **Search** box at the top right of the documentation home page:

http://www.iona.com/support/docs

To search a particular library version, browse to the required index page, and use the **Search** box at the top right, for example:

http://www.iona.com/support/docs/artix/mainframe/5.1/index.xml

You can also search within a particular book. To search within a HTML version of a book, use the **Search** box at the top left of the page. To search within a PDF version of a book, in Adobe Acrobat, select **Edit**|**Find**, and enter your search text.

## Artix Online Help

Artix Designer and Artix Orchestration Designer include comprehensive online help, providing:

- Step-by-step instructions on how to perform important tasks
- A full search feature
- Context-sensitive help for each screen

There are two ways that you can access the online help:

- Select **Help|Help Contents** from the menu bar. The help appears in the contents panel of the Eclipse help browser.
- Press **F1** for context-sensitive help.

In addition, there are a number of cheat sheets that guide you through the most important functionality in Artix Designer and Artix Orchestration Designer. To access these, select **Help|Cheat Sheets**.

## Artix Glossary

The Artix Glossary is a comprehensive reference of Artix terms. It provides quick definitions of the main Artix components and concepts. All terms are defined in the context of the development and deployment of Web services using Artix.

## Additional Resources

The IONA Knowledge Base (http://www.iona.com/support/knowledge_base/index.xml) contains helpful articles written by IONA experts about Artix and other products.

The IONA Update Center (http://www.iona.com/support/updates/index.xml) contains the latest releases and patches for IONA products.

If you need help with this or any other IONA product, go to IONA Online Support (http://www.iona.com/support/index.xml).

Comments, corrections, and suggestions on IONA documentation can be sent to docs-support@iona.com .

## Document Conventions

### Typographical conventions

This book uses the following typographical conventions:

`Fixed width`

Fixed width (Courier font) in normal text represents portions of code and literal names of items such as classes, functions, variables, and data structures. For example, text might refer to the `IT_Bus::AnyType` class.

Constant width paragraphs represent code examples or information a system displays on the screen. For example:

```
#include <stdio.h>
```

*`Fixed width italic`*

Fixed width italic words or characters in code and commands represent variable values you must supply, such as arguments to commands or path names for your particular system. For example:

```
% cd /users/YourUserName
```

*Italic*

Italic words in normal text represent *emphasis* and introduce *new terms*.

**Bold** Bold words in normal text represent graphical user interface components such as menu commands and dialog boxes. For example: the **User Preferences** dialog.

**Keying Conventions**

This book uses the following keying conventions:

No prompt When a command's format is the same for multiple platforms, the command prompt is not shown.

`%` A percent sign represents the UNIX command shell prompt for a command that does not require root privileges.

`#` A number sign represents the UNIX command shell prompt for a command that requires root privileges.

`>` The notation > represents the MS-DOS or Windows command prompt.

```
. . .
.
.
.
```

Horizontal or vertical ellipses in format and syntax descriptions indicate that material has been eliminated to simplify a discussion.

`[ ]` Brackets enclose optional items in format and syntax descriptions.

`{ }` Braces enclose a list from which you must choose an item in format and syntax descriptions.

`|` In format and syntax descriptions, a vertical bar separates items in a list of choices enclosed in `{ }` (braces).

In graphical user interface descriptions, a vertical bar separates menu commands (for example, select **File**|**Open**).

# Installing Artix Mainframe

**Overview**

This section introduces the Artix Mainframe installation procedure and lists the supported platforms and compilers.

**Before you begin**

This guide describes how to install the mainframe components of Artix Mainframe.

The off-host components of Artix Mainframe are installed with Artix 5.1 for Windows, Linux, and Solaris platforms. See the *Artix ESB Installation Guide* for information on installing these components.

Before you install the mainframe components:

- Confirm that you are in compliance with the prerequisites for installing Artix Mainframe, as described in "Installation Prerequisites" on page 15.
- Familiarize yourself with the steps involved in installing this product.

**License code**

To run an Artix application, you must have a valid license code from IONA Technologies for the Artix components that will interoperate. For example, if you plan to connect Windows clients to a z/OS-based service, you must have license codes for the Windows and z/OS components. If you plan to connect z/OS clients to a Linux-based service, you must have license codes for the Linux and z/OS components.

License codes are delivered from IONA by e-mail. *If you do not have valid license codes, please contact IONA support at support@iona.com or contact your IONA account representative before you proceed*.

**Two-phase installation**

Installation of Artix Mainframe is performed in two phases:

- Installing the z/OS-based components (described in this manual).
- Installing the off-host components (described in the *Artix ESB Installation Guide*). The off-host components of Artix Mainframe are installed with Artix for Windows, Linux, and Solaris platforms.

You can install the z/OS-based components independently of the off-host components. There is no requirement for either set of components to be installed first.

# Supported Platforms

### Overview

This section describes the z/OS, CICS, IMS, COBOL, PL/I, WebSphere MQ, and DB2 versions supported by Artix Mainframe.

### Supported platforms

The supported platforms for the Windows and Linux components of Artix are described in the *Artix ESB Installation Guide.*

The supported platforms for the z/OS-based components are:

- z/OS V1R7
- z/OS V1R8
- z/OS V1R9

### Supported CICS releases

The supported CICS releases are:

- CICS TS V2.2
- CICS TS V2.3
- CICS TS V3.1
- CICS TS V3.2

**Note:** The BMS parser component of Artix requires CICS TS V2.2 or later.

**Note:** The ability to send messages greater than 32K into CICS requires CICS TS V3.1 or later.

### Supported IMS releases

The supported IMS releases are:

- IMS for z/OS V8.1
- IMS for z/OS V9.1

### Supported COBOL compilers

**Exposing existing applications as Web services or CORBA objects**

Artix Designer and the command-line utilities of Artix Mainframe both offer support for parsing copybooks from CICS programs and IMS transactions that have been built with the following COBOL compilers on z/OS:

- IBM COBOL for z/OS V2.2.1

- IBM Enterprise COBOL V3.4.1
- IBM VS COBOL II

**Creating Web service applications from WSDL**

Artix Designer and the command-line utilities of Artix Mainframe both offer support for generating COBOL copybooks for use in CICS programs and IMS transactions. These copybooks build with the following compilers on z/OS:

- IBM Enterprise COBOL V3.4.1

## Supported PL/I compilers

Artix Designer and the command-line utilities of Artix Mainframe both offer support for generating PL/I include files for use in CICS programs and IMS transactions. These include files build with the following compilers on z/OS:

- IBM PL/I for MVS and VM V1.1.1
- IBM Enterprise PL/I V3.4.0, V3.5.0, or V3.6.0

## Supported WebSphere MQ releases

The supported WebSphere MQ releases are:

- IBM WebSphere MQ for z/OS V5.3.1
- IBM WebSphere MQ for z/OS V6.0

## Supported DB2 release

The supported release of DB2 is IBM Universal Database for z/OS V8.

# Installation Prerequisites

### Overview

This section describes the prerequisites for installing the mainframe components of Artix Mainframe.

### z/OS System Requirements

The following basic program temporary fixes (PTFs) are required:

**Note:** Check http://www.iona.com/support/docs/apars/index.xml for the PTF details and for a more up-to-date list of IBM maintenance requirements for IONA products.

| Operating System | Required Patches |
|---|---|
| z/OS V1R7 | UA34180, UA23848, UK08059, UK08244, UK09123, UK09695, UK10244 |
| z/OS V1R8 | UA34181, UA32974, UK19837, UK21554, UK21780, UK23297 |
| z/OS V1R9 | UA34182 |

The following PTFs are also required if you wish to use SSL with Artix:

| Operating System | Required Patches |
|---|---|
| z/OS V1R7 | UA23758 |
| z/OS V1R8 | There are no PTF requirements. |
| z/OS V1R9 | There are no PTF requirements. |

## IMS requirements

The following PTFs are required for OTMA If you wish to use IMS with Artix:

| IMS Version | Required OTMA Patches |
|---|---|
| IMS V8 | UK03269, UQ63252, UQ63672, UQ69205, UQ79301, UQ79902, UQ82806, UQ82807, UQ88712, UQ94715 |
| IMS V9 | UK03271, UQ91993 |

The following PTFs are required for Resource Recovery Service (RRS) if you wish to use IMS with Artix:

| IMS Version | Required RRS Patches |
|---|---|
| IMS V8 | UK09100, UQ60227, UQ63218, UQ69204, UQ89956 |
| IMS V9 | UK09099, UQ91845 |

## CICS requirements

The following PTFs are required if you wish to use CICS with Artix:

| CICS Version | Required Patches |
|---|---|
| CICS TS V3.2 | OA19565 |

## WebSphere MQ requirements

The following PTF is required if you wish to use IBM WebSphere MQ:

| WebSphere MQ Version | Required Patch |
|---|---|
| WebSphere MQ V5.3.1 | UK11380 |

## DB2 requirements

There are currently no PTF requirements for IBM Universal Database for z/OS.

## z/OS disk space requirements

The approximate amount of disk space required to install the z/OS-based component of Artix is:

| Files | Space |
|---|---|
| ARTIX sequential file | 150 3390-3 cylinders |
| Unpacked sequential file (PDS) | 153 3390-3 cylinders |
| Product as installed | 294 3390-3 cylinders |

## Runtime environment requirements

The following runtime environment requirements apply:

| Prerequisite | Notes |
|---|---|
| C++ Runtime Libraries | The IBM Language Environment (SCEERUN) and C++ runtime libraries (SCLBDLL) must be available when running any Artix service. |

## SSL requirements

By default, SSL is configured to use 128-bit (high strength) encryption. 128-bit encryption requires that one of the following IBM System SSL V3 FMIDs is installed in your operating system environment:

| Operating System | FMID |
|---|---|
| z/OS V1R7 | JCPT361 |
| z/OS V1R8 | JCPT381 |
| z/OS V1R9 | JCPT391 |

If these FMIDs are unavailable, SSL can be configured to use weaker encryption. See “SSL Customization Tasks” on page 73 for further information.

The following requirements apply if you plan to run services or programs with SSL enabled:

- To run the supplied GENCERT JCL, which sets up the various keyrings, you must be authorized to issue the RACDCERT CERTAUTH command. The authority to issue this command is controlled by having CONTROL access to the IRR.DIGTCERT.function resource in the FACILITY class.

  **Note:** Although having READ and UPDATE access to the IRR.DIGTCERT.function resource grants authority to issue the RACDCERT command within certain limits, you must have CONTROL access to the IRR.DIGTCERT.function, because the supplied GENCERT and DELCERT JCL members respectively create and delete sample CERTAUTH certificates.

  For detailed information about the RACDCERT command, and the authority required to execute each operand, see the IBM publication: *z/OS Security Server (RACF) Command Language Reference*.

  For more details about how to configure the transformer service to run in secure mode, see "SSL Customization Tasks" on page 73.

- Ensure that the RACF DIGTCERT and DIGTRING general resource classes have been activated. If not, ask your RACF administrator to issue the following commands:

  ```
  SETROPTS CLASSACT(DIGTCERT)
  SETROPTS CLASSACT(DIGTRING)
  ```

- IBM strongly recommends that you issue the RACLIST command on the DIGTCERT class, to improve performance when using digital certificates. If you do not issue the RACLIST command on the DIGTCERT class, digital certificates can still be used, but performance might be affected. For best performance, issue the following command:

  ```
  SETROPTS RACLIST(DIGTCERT)
  ```

- After creating a new digital certificate, refresh the DIGTCERT class by issuing the following command:

  ```
  SETROPTS RACLIST(DIGTCERT) REFRESH
  ```

> If you do not refresh the `DIGTCERT` profiles on which the `RACLIST` command has been issued, RACF still uses the new digital certificate, but performance might be affected.

For more information about creating keyrings and storing digital certificates in RACF, see the IBM publication: *z/OS Security Server (RACF) Security Administrator's Guide*.

## Kerberos authentication requirements

Artix Mainframe supports the validation of Kerberos tokens sent to it from off-host Web service clients, using either RACF or an off-host iS2 server.

Before Kerberos authentication can be used with Artix, a number of steps to enable the Network Authentication Service are required on your z/OS system. Network Authentication Service is a component of IBM's z/OS Security Server and is IBM's implementation of Kerberos Version 5 from Massachusetts Institute of Technology (MIT).

To configure Network Authentication Service on your z/OS system, follow the instructions in the section “Making the program operational” in the IBM publication: *z/OS Integrated Security Services Network Authentication Service Administration - SC24-5926*. Depending on your installation, one or all of these tasks might already have been completed. When complete, you will have the SKRBKDC started-task running on your z/OS system, with a registry database defined and the required RACF definitions in place.

## Artix Security Service integration requirements

Artix Mainframe can be fully integrated with an Artix Security Service server running off-host. If you wish to use this feature, you will need access to the XML4C Version 5.4 (Xerces C++ Version 2.4) XML parser on z/OS. This XML parser is available in the XML Toolkit V1.7, which can be downloaded free of charge from the IBM website (http://www-03.ibm.com/servers/eserver/zseries/software/xml/), or you can order it through the normal IBM channels. The runtime DLLs must be made available in the *native* environment for use by the transformer service running as a batch job or started task.

## ARXVERSN utility requirements

To run the ARXVERSN utility that is supplied with Artix Mainframe, the IXM4C54 definition load module must be available in the job region. Otherwise, an error will be reported.

If you wish to run the ARXVERSN utility, you will need access to the XML4C Version 5.4 (Xerces C++ Version 2.4) XML parser on z/OS. This XML parser can be downloaded free of charge from the IBM web site at :http://www-03.ibm.com/servers/eserver/zseries/software/xml/. Alternatively, you can order it through the normal IBM channels.

Once this product is installed, you must make the IXM.SIXMLOD1 load library available to the ARXVERSN job by updating *artixhlq*.PROCLIB(ORXG). This adds the XML Toolkit Loadlib library to your STEPLIB concatenation. The relevant line is shipped for your convenience as a comment, as follows:

```
//*   DD DISP=SHR,DSN=&XMLLOAD
```

## XML parsing with XML System Services

Artix Mainframe can be configured to perform some XML parsing using IBM's XML System Services. XML System Services is a set of callable services that are part of z/OS. The callable services provide facilities for parsing XML documents. For details on how to configure Artix Mainframe to use XML System Services see the Artix Mainframe Administrator's Guide.

The following PTFs are required if you want to use XML System Services:

| Operating System | Required Patches |
|---|---|
| z/OS 1.7 | UA31443 |

**Note:** In z/OS 1.8 and higher versions, XML System Services is integrated into the operating system and no PTFs are required to make the services available.

# Installing Artix Mainframe Components on z/OS

**Overview**

This section explains the steps to install the z/OS-based components of Artix Mainframe.

This release includes an `ARTIX.SEQ` file for installing the z/OS-based components, which is shipped as an IEBCOPY backup file that has been compressed using the TSO `XMIT` command.

**Note:** Read each step in full before proceeding with it, because the text might contain important recommendations or requirements that you need to know before proceeding.

**In this section**

This section discusses the following topics:

## Before You Begin Installing

**Overview**

This subsection underscores important pre-installation information.

**Customizing the product**

After you have successfully installed the product, you must perform some customization tasks before you can actually use it. These customization tasks are described in "Customizing Your Artix Mainframe Installation" on page 32.

**Sequence of Tasks**

Do not attempt to perform any installation or customization tasks out of sequence. Installation must be successfully completed before you begin customization. Perform all tasks in the order in which they are described in this guide.

# Installation Steps

### Overview

This subsection describes how to install the z/OS-based components of Artix Mainframe.

**Note:** You must complete all the steps in this subsection in the order in which they are presented.

### Step 1: Preallocate a data set

Preallocate a z/OS sequential data set with the following information:

| Space Units | Tracks |
|---|---|
| Primary quantity | 2300 |
| Secondary quantity | 100 |
| Directory blocks | 0 |
| Record format | FB |
| Record length | 80 |
| Block size | 3120 |

### Step 2: Copy the ARTIX.SEQ file

Copy the `ARTIX.SEQ` file from your product CD into the z/OS data set that you preallocated in the preceding step. How you copy the file depends on the type of machine the CD-ROM drive is on. The most convenient way is to use FTP.

The following is an example of the FTP command sequence to transmit the `ARTIX.SEQ` file into the preallocated data set, where the CD drive letter is `d:` and *`XXXX.XXXX`* represents the name of the data set:

```
d:
ftp zOShost
ftp> binary
ftp> put ARTIX.SEQ 'XXXX.XXXX'
```

### Step 3: Unpack the PDS

After the `ARTIX.SEQ` file has been copied to z/OS, use the TSO `RECEIVE` command to unpack the PDS (where *`XXXX.XXXX`* represents the exact name of the PDS data set that is to be received):

```
RECEIVE INDSN('XXXX.XXXX')
```

Because the preceding command is a TSO command, you must enter it on an ISPF command screen.

You are prompted with restore parameters similar to the following:

```
To receive the Artix Mainframe PDS, please specify the following:
DA('HLQ.ARTIX51.PDS') SPACE(2870,100) REL
replacing the HLQ as appropriate.
INMR901I Dataset HLQ.ARTIX51.PDS from BUILD on NODENAME
INMR906A Enter restore parameters or 'DELETE' or 'END' +
```

You must choose between one of the following:

- Press **Enter**, to restore *`XXXX.XXXX`* into the default data set, `HLQ.ARTIX51.PDS`.
- Restore *`XXXX.XXXX`* into an alternative data set, by entering the command that appears on your screen and substituting `HLQ.ARTIX51.PDS` with the data set name you want to use.

The sequential data set, *`XXXX.XXXX`*, can now be deleted.

### Step 4: Expand the PDS

The *`artixhlq`*`.PDS($FIRST)` member contains JCL to expand the other PDS members into the full Artix PDS installation. The default high-level qualifier for installation data sets is `HLQ.ARTIX51`. If you want to change the default high-level qualifier to your installation standard, edit the `$FIRST` member, using the following command in ISPF:

```
C 'HLQ.ARTIX51'   'artixhlq' ALL
```

In the preceding example, *`artixhlq`* represents your high-level qualifier, which can be up to 21 characters, including one or more periods.

Now submit *`artixhlq`*`.PDS($FIRST)` to install Artix PDS.

**Note:** This step might take several minutes to complete.

### Step 5: Customize the Artix default installation HLQ

The default high-level qualifier used throughout the Artix package (that is, in JCL members, PROCS, readmes, and configuration files) is `HLQ.ARTIX51`. In each case, this high-level qualifier must be changed to match the high-level qualifier that you used for your installation when you submitted the `$FIRST` job. The *`artixhlq`*`.PDS($SECOND)` member contains JCL to convert all the references of `HLQ.ARTIX51` in the Artix package to match your high-level qualifier. To enable the `$SECOND` job to do this, perform the following steps:

1. Edit the *`artixhlq`*`.PDS($SECOND)` member, using the following command in ISPF:

   ```
   C 'INSTALHLQ'   'artixhlq' ALL
   ```

   In the preceding command, *`artixhlq`* must match the high-level qualifier you specified in the `$FIRST` job in “Step 4: Expand the PDS” on page 24.

2. Submit `$SECOND` to convert all the references of `HLQ.ARTIX51` in the Artix package to match your high-level qualifier.

**Note:** This step might take a couple of minutes to complete.

### Step 6: Customize your locale (if necessary)

*This is only relevant if you want to run Artix in a locale other than the default locale IBM-1047, and your system and compiler are also running in a locale other than IBM-1047.*

Artix include files and demonstration sources are coded by default in locale IBM-1047. Follow these steps if you do not want to run Artix in the default IBM-1047 locale, and your system and compiler are also running in a locale other than IBM-1047:

1. In *`artixhlq`*`.PDS($THIRD)`, use the following command in ISPF to change the default high-level qualifier, to make it match your installation value (where *`artixhlq`* represents your high-level qualifier, which can be up to 21 characters, including one or more periods):

   ```
   C 'HLQ.ARTIX51' 'artixhlq' ALL
   ```

2. In *artixhlq*.PDS, use the following command in ISPF to change the value of the TO variable, to make it match the locale codeset you want to use (where IBM-*xxx* represents your codeset):

```
C 'IBM-500' 'IBM-xxx' ALL
```

   The preceding command lets you simultaneously change all occurrences of the default to make it match your codeset.

3. Submit $THIRD to convert the files to match your installation.

### Step 7: Check installed data sets

Compare your list of installed data sets with the list shown in Table 1.

**Table 1:** *Data sets installed with Artix Mainframe*

| Data Set | Description |
|---|---|
| *artixhlq*.BATCH.CLNT.BRGEINFO | Used to store the XML-based type information file for Artix clients, created by Artix Designer for batch applications. |
| *artixhlq*.BATCH.CLNT.BRGEMAP | Used to store the mapping file for Artix clients, created by Artix Designer for batch applications. |
| *artixhlq*.CBL.OBJLIB | Contains client-side API library for batch and IMS COBOL applications. |
| *artixhlq*.CICS.CBL.OBJLIB | Contains client-side API library for CICS COBOL applications. |
| *artixhlq*.CICS.CLNT.BRGEINFO | Used to store the XML-based type information file for Artix clients, created by Artix Designer for CICS applications. |
| *artixhlq*.CICS.CLNT.BRGEMAP | Used to store the mapping file for Artix clients created by Artix Designer for CICS applications. |
| *artixhlq*.CICS.PLI.OBJLIB | Contains client-side API library for CICS PL/I applications. |

**Table 1:** *Data sets installed with Artix Mainframe*

| Data Set | Description |
|---|---|
| `*artixhlq*.CICS.SRVR.BRGEINFO` | Used to store the XML-based type information file for Artix servers, created by Artix Designer for CICS applications. |
| `*artixhlq*.CICS.SRVR.BRGEMAP` | Used to store the mapping file for Artix servers created by Artix Designer for CICS applications. |
| `*artixhlq*.CONFIG` | Contains Artix configuration information. |
| `*artixhlq*.DEMOS.CBL.COPYLIB` | Used to store copybooks for the COBOL Batch demonstrations. |
| `*artixhlq*.DEMOS.CBL.JCLLIB` | Contains jobs to build and run the COBOL Batch demonstrations. |
| `*artixhlq*.DEMOS.CBL.LOADLIB` | Used to store the load modules for the COBOL Batch demonstrations. |
| `*artixhlq*.DEMOS.CBL.README` | Contains instructions for building and running the supplied COBOL Batch demonstrations. |
| `*artixhlq*.DEMOS.CBL.SRC` | Contains program source for the COBOL Batch demonstrations. |
| `*artixhlq*.DEMOS.CICS.CBL.COPYLIB` | Used to store copybooks for the COBOL CICS demonstrations. |
| `*artixhlq*.DEMOS.CICS.CBL.JCLLIB` | Contains jobs to build and run the COBOL CICS demonstrations. |
| `*artixhlq*.DEMOS.CICS.CBL.LOADLIB` | Used to store the load modules for the COBOL CICS demonstrations. |
| `*artixhlq*.DEMOS.CICS.CBL.README` | Contains instructions for building and running the supplied COBOL CICS demonstrations. |

**Table 1:** *Data sets installed with Artix Mainframe*

| Data Set | Description |
|---|---|
| *artixhlq*.DEMOS.CICS.CBL.SRC | Contains program source for the COBOL CICS demonstrations. |
| *artixhlq*.DEMOS.CICS.PLI.JCLLIB | Contains jobs to build and run the PL/I CICS demonstrations. |
| *artixhlq*.DEMOS.CICS.PLI.LOADLIB | Used to store the load modules for the PL/I CICS demonstrations. |
| *artixhlq*.DEMOS.CICS.PLI.PLINCL | Used to store copybooks for the PL/I CICS demonstrations. |
| *artixhlq*.DEMOS.CICS.PLI.README | Contains instructions for building and running the supplied PL/I CICS demonstrations. |
| *artixhlq*.DEMOS.CICS.PLI.SRC | Contains program source for the PL/I CICS demonstrations. |
| *artixhlq*.DEMOS.DB2.STATMNTS | This file is used to store the SQL statements added to the DB2 gateway by the UI. |
| *artixhlq*.DEMOS.IDL | Used to store IDL for the CORBA server demonstrations. |
| *artixhlq*.DEMOS.IMS.CBL.COPYLIB | Used to store copybooks for the COBOL IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.CBL.JCLLIB | Contains jobs to build and run the COBOL IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.CBL.LOADLIB | Used to store the load modules for the COBOL IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.CBL.README | Contains instructions for building and running the supplied COBOL IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.CBL.SRC | Contains program source for the COBOL IMS demonstrations. |

**Table 1:** *Data sets installed with Artix Mainframe*

| Data Set | Description |
|---|---|
| *artixhlq*.DEMOS.IMS.JCLLIB | Contains jobs to build and run the IMS MFS demonstration. |
| *artixhlq*.DEMOS.IMS.PLI.JCLLIB | Contains jobs to build and run the PL/I IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.PLI.LOADLIB | Used to store the load modules for the PL/I IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.PLI.PLINCL | Used to store include files for the PL/I IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.PLI.README | Contains instructions for building and running the supplied PL/I IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.PLI.SRC | Contains program source for the PL/I IMS demonstrations. |
| *artixhlq*.DEMOS.IMS.README | Contains instructions for building and running the IMS MFS demonstration. |
| *artixhlq*.DEMOS.PLI.JCLLIB | Contains jobs to build and run the PL/I Batch demonstrations. |
| *artixhlq*.DEMOS.PLI.LOADLIB | Used to store the load modules for the PL/I Batch demonstrations. |
| *artixhlq*.DEMOS.PLI.PLINCL | Used to store include files for the PL/I Batch demonstrations. |
| *artixhlq*.DEMOS.PLI.README | Contains instructions for building and running the supplied PL/I Batch demonstrations. |
| *artixhlq*.DEMOS.PLI.SRC | Contains program source for the PL/I Batch demonstrations. |
| *artixhlq*.DOC | Contains miscellaneous documentation. |

**Table 1:** *Data sets installed with Artix Mainframe*

| Data Set | Description |
|---|---|
| *artixhlq*.DOC.IMAGES | Contains images used by the WSDL browsing feature. |
| *artixhlq*.IMS.CLNT.BRGEINFO | Used to store the XML-based type information file for Artix clients, created by Artix Designer for IMS applications. |
| *artixhlq*.IMS.CLNT.BRGEMAP | Used to store the mapping file for Artix clients, created by Artix Designer for IMS applications. |
| *artixhlq*.IMS.SRVR.BRGEINFO | Used to store the XML-based type information file for Artix servers, created by Artix Designer for IMS applications. |
| *artixhlq*.IMS.SRVR.BRGEMAP | Used to store the mapping file for Artix servers, created by Artix Designer for IMS applications. |
| *artixhlq*.INCLUDE.COPYLIB | Contains include files for COBOL demonstration programs. |
| *artixhlq*.INCLUDE.PLINCL | Contains include files for PL/I demonstration programs. |
| *artixhlq*.IORS | Used to store IORs for the CORBA server demonstrations. |
| *artixhlq*.JCLLIB | Contains jobs to run Artix. |
| *artixhlq*.LKED | Contains side-decks for the DLLs. |
| *artixhlq*.LOADLIB | Contains binaries and DLLs for the transformer service. |
| *artixhlq*.MFA.LOADLIB | Contains run-time DLLs for IMS and CICS client applications. |
| *artixhlq*.PLI.OBJLIB | Contains client-side API library for batch and IMS PL/I applications. |

**Table 1:** *Data sets installed with Artix Mainframe*

| Data Set | Description |
|---|---|
| `artixhlq.PROCLIB` | Contains JCL procedures. |
| `artixhlq.REXX` | Contains the source code for performing the `HLQ.ARTIX51` to `dest-install-hlq` customization step. |

# Customizing Your Artix Mainframe Installation

**Overview**

This section describes the customization tasks to be performed on z/OS after installing the z/OS-based components of Artix Mainframe. You must customize your Artix Mainframe installation before you can use it.

This section first describes the standard tasks that you must perform, and then describes additional customization that you might need to perform, depending on your setup.

**Note:** Read each step in full before proceeding with it, because the text might contain important recommendations or requirements that you need to know before proceeding.

**References to Artix Mainframe documentation**

Some customization steps refer you to detailed explanations in the Artix Mainframe 5.1 documentation, which is available at http://www.iona.com/support/docs/artix/mainframe/5.1/index.xml.

**In this section**

This section discusses the following topics:

# Standard Customization Tasks

**Overview**

This subsection describes standard customization tasks that you must perform before you can use Artix Mainframe. You must perform these customization tasks in the order in which they are presented.

**Step 1: Verify change of data set name defaults in ARTXVARS**

Verify that the default high-level qualifier in `*artixhlq*.PROCLIB(ARTXVARS)` has been successfully changed to the proper value for your installation. This change should have been automatically performed when you ran the `$SECOND` job in "Step 5: Customize the Artix default installation HLQ" on page 25.

**Step 2: Verify variable settings**

Verify that the following variables in `*artixhlq*.PROCLIB(ARTXVARS)`, which represent system data set high-level qualifiers, match those installed on your z/OS system:

**Table 2:** *ARTXVARS variables to verify*

| | |
|---|---|
| `TCPIP` | This is the high-level qualifier for the IBM TCP/IP SEZARNT1 and SEZACMTX libraries. For example:<br>`SET TCPIP=TCPIP` |
| `TCPIPCFG` | This is the TCP/IP configuration file to be used. It is the file referred to as the TCPIP.DATA file in the IBM TCP/IP publications. For example:<br>`SET TCPIPCFG=SYS1.TCPPARMS(TCPDATA)` |
| `CEE` | This is the high-level qualifier for the IBM Language Environment (L/E) C data sets, such as the SCEELKED library needed to link the sample demonstrations. For example:<br>`SET CEE=CEE`<br>This entry is used to control the location of the `CEERUN`, `CEEMSGP`, `CEECPP`, and `CEELKED` LE data sets. If you have customized any of these data sets, you might need to update these entries immediately following `CEE`. |

**Table 2:** *ARTXVARS variables to verify*

| | |
|---|---|
| `CBC` | This is the high-level qualifier for the IBM C++ compiler data sets, such as the SCLBDLL library. For example:<br>`SET CBC=CBC`<br>This entry is used to control the location of the CBCSID and CDCDLL C++ data sets. If you have customized any of these data sets, you might need to update these entries immediately following `CBC`. |
| `CICSHLQ` | This is the high-level qualifier for the IBM CICS libraries. For example:<br>`SET CICSHLQ=CICSTS22`<br>This entry is used to control the location of the `CICSMAC`, `CICSLOAD`, `CICSCOB`, and `CICSEXCI` CICS data sets. If you have customized any of these data sets, you might need to updates these entries immediately following `CICSHLQ`. |
| `IMSHLQ` | This is the high-level qualifier for the IBM IMS libraries. For example:<br>`SET IMSHLQ=IMS810`<br>This entry is used to control the location of the IMSRES and IMSISRC IMS data sets. If you have customized any of these data sets, you might need to update these entries immediately following `IMSHLQ`. |
| `CBLPRFX` | This is the high-level qualifier for the IBM COBOL libraries needed to link the sample demonstrations. For example:<br>`SET CBLPRFX=IGY` |
| `PLIPRFX` | Determines the PL/I procedure that is executed.<br>• IEL1 will execute the IEL1C procedure, which invokes the PL/I for MVS compiler.<br>• IBMZ will execute the IBMZC procedure, which invokes the Enterprise PL/I compiler.<br>For example:<br>`SET PLIPRFX=IEL1` |

**Table 2:** *ARTXVARS variables to verify*

| | |
|---|---|
| `PLNGPRFX` | This is used to specify the PL/I LNGPRFX compiler option and works in conjunction with PLIPRFX. This must be set to the high-level qualifier of the PL/I compiler library.<br>For example, the following settings will pick up the enterprise PL/I procedure, IBMZC, and the enterprise PL/I compiler from the IBMZ.SIBMZCMP library:<br>`SET PLIPRFX=IBMZ`<br>`SET PLNGPRFX=IBMZ`<br>For example, the following settings will pick up the enterprise PL/I procedure, IEL1C, and the PL/I for MVS compiler from the IEL.SIELCOMP library:<br>`SET PLIPRFX=IEL1`<br>`SET PLNGPRFX=IEL`<br>**Note:** The default is `IEL`. |
| `MQHLQ` | This is the high-level qualifier for the IBM WebSphere MQ libraries. For example:<br>`SET MQHLQ=MQM`<br>This entry is used to control the location of the `MQLOAD`, `MQAUTH`, and `MQLANG` MQ data sets. If you have customized any of these data sets, you might need to update these entries immediately following `MQHLQ`. |
| `DB2HLQ` | This is the high-level qualifier for the IBM DB2 libraries. For example:<br>`SET DB2HLQ=DSN810`<br>This entry is used to control the location of the `DB2LOAD` and `DB2EXIT` DB2 data sets. If you have customized any of these data sets, you might need to update these entries immediately following `DB2HLQ`. |

**Table 2:** *ARTXVARS variables to verify*

| | |
|---|---|
| `XMLHLQ` | This is the high-level qualifier for the IBM XML toolkit. For example:<br>`SET XMLHLQ=IXM`<br>This entry is used to control the location of the `XMLLOAD` XML data set. If you have customized this data set, you might need to update this entry immediately following `XMLHLQ`. |
| `SSLHLQ` | This is the high-level qualifier for the IBM System SSL load library. For example:<br>`SET SSLHLQ=SYS1`<br>This entry is used to control the location of the `SSLLOAD` SSL data set. If you have customized this data set, you might need to update this entry immediately following `SSLHLQ`. |

If the supplied defaults do not match those in use at your site, change them as appropriate.

### Step 3: Set ITTIMEZ (if necessary)

*The timestamps displayed in the Artix logging output are based on the time zone as configured on your system. If you would like your Artix applications to display timestamps in an alternate time zone, you may use the IONA ITTIMEZ variable to override the system setting.*

The `ITTIMEZ` variable in `*artixhlq*.PROCLIB(ARTXVARS)` file provides a convenient way to use the LE TZ setting to override the default system time zone. For example, if your system time is configured for GMT, but you would like your Artix Mainframe logs to be displayed in central European time (GMT+1) taking into account daylight savings time, set the ITTIMEZ as follows:

```
SET ITTIMEZ='TZ=”GMT1GDT”'
```

Similarly, to have Artix run in the American EST time zone, set ITTIMEZ as follows:

```
SET ITTIMEZ='TZ=”EST5EDT”'
```

### Step 4: Set ITLOCALE (if necessary)

*This is only relevant if you want to run Artix in a locale other than IBM-1047, and your system and compiler are running in a locale other than the locale in which you want to run Artix.*

If you plan to run Artix in a locale other than IBM-1047, and your system and compiler are running in a locale other than the locale in which you want to run Artix, set the `ITLOCALE` variable in `*artixhlq*.PROCLIB(ARTXVARS)` to the locale in which you want to run Artix. For example, to have Artix run in the Swiss German locale, set ITLOCALE as follows:

```
SET ITLOCALE='LC_ALL=De_CH.IBM-500'
```

If you have set the ITTIMEZ variable, you must include a comma before the ITLOCALE setting as follows:

```
SET ITLOCALE=',LC_ALL=De_CH.IBM-500'
```

See the “Passing Program Parameters” section of the Artix Mainframe *Administrator’s Guide* for more details of alternative methods of passing program parameters in Artix Mainframe.

## Step 5: Set ITENVAR (if necessary)

The ITTIMEZ and ITLOCALE settings are used to create an ITENVAR variable that forms the ENVAR statement, which is passed as an RPARM to all Artix jobs. For example:

```
SET ITENVAR=&Q,ENVAR(&ITTIMEZ&ITLOCALE)&Q
```

If you have set either the ITTIMEZ or ITLOCALE configuration item, you must remove the comment character that precedes the ITENVAR setting, because it is not enabled by default.

**Note:** Depending on the size of the other program parameters and runtime parameters that are being passed, you might exceed the 100-byte limitation and receive the following JCL error:

```
43 IEF6421 EXCESSIVE PARAMETER LENGTH IN THE PARM FIELD
```

If you receive the preceding JCL error, you must reduce the length of the parameters being passed. You can use the ORBARGS DD card to pass some of the Artix-specific program parameters. For more details see the Artix Mainframe *Administrator's Guide*.

## Step 6: Set up your license file

The product license information that you have received by e-mail needs to be transferred to the mainframe and formatted before it can be used by Artix Mainframe. Follow these steps:

1. Preallocate a z/OS sequential data set on the host with the following information:

| Space Units | Tracks |
|---|---|
| Primary quantity | 1 |
| Secondary quantity | 1 |
| Directory blocks | 0 |
| Record format | VB |
| Record length | 500 (or greater) |
| Block size | 0 |

2. Use FTP to transfer the license as a text file into the newly created data set. The following is an example of the FTP command sequence, where the drive letter is `C:` and *`XXXX.XXXX`* represents the name of the data set you have just allocated:

```
C:
ftp zOShost
ftp> asc
ftp> put license.txt 'XXXX.XXXX'
```

3. After the license text file has been copied to z/OS, edit the JCL in *`artixhlq`*`.JCLLIB(ARXCOPY)`, as follows:

   - Change the default high-level qualifier to reflect the proper value for your installation. You can use the following command in ISPF to do this (where *`artixhlq`* represents your high-level qualifier, which can be up to 21 characters):

     ```
     C 'HLQ.ARTIX51' 'artixhlq' ALL
     ```

   - On the `IN DD` statement, replace where it says *`your VB data set here`* with the name of the data set that contains your license file.

4. Submit `ARXCOPY` to copy the license file to *`artixhlq`*`.CONFIG(LICENSES)`. The `ARXCOPY` job copies the license file from a variable-length record file into the fixed-length record license file used by Artix. It splits long lines across records, delimiting them with a backslash in column 72.

---

### Step 7: Convert your license file (if necessary)

*This is only relevant if you want to run Artix in a locale other than the default locale, IBM-1047.*

If so, the steps are:

1. In *`artixhlq`*`.PDS($FOURTH)`, use the following command in ISPF to change the default high-level qualifier, to make it match your installation value (where *`artixhlq`* represents your high-level qualifier, which can be up to 19 characters including one or more periods):

   ```
   C 'HLQ.ARTIX51' 'artixhlq' ALL
   ```

2. In *`artixhlq`*`.PDS($FOURTH)`, use the following command in ISPF to change the value of the TO variable, to make it match the locale codeset in which you want to run Artix (where `IBM-`*`xxx`* represents the codeset):

   ```
   C 'IBM-500' 'IBM-xxx' ALL
   ```

   The preceding command lets you simultaneously change all occurrences of the default to make it match your codeset.

3. Submit *`artixhlq`*`.PDS($FOURTH)` to convert your license file.

---

### Step 8: Update your configuration file

Before you can use any of the supplied Artix services, values must be assigned to the following configuration variable in *`artixhlq`*`.CONFIG(ARTIX)`:

`LOCAL_HOSTNAME` This specifies the fully qualified local hostname.

### Step 9: Confirm availability of default ports

Verify that the ports used by default for the services you are implementing in the standard configuration file are available on your system. If not, assign available values in *artixhlq*.CONFIG(ARTIX). The eight default values are listed in the following table and are described in the Artix Mainframe *Administrator's Guide*.

| | |
|---|---|
| HTTP_BATCH_CLIENT_PORT | Default value is 12030. |
| HTTP_IMS_CLIENT_PORT | Default value is 12040. |
| HTTP_IMS_SERVER_PORT | Default value is 12050. |
| HTTP_CICS_CLIENT_PORT | Default value is 12041. |
| HTTP_CICS_SERVER_PORT | Default value is 12051. |
| HTTP_DB2_SERVER_PORT | Default value is 12500. |
| IIOP_IMS_SERVER_PORT | Default value is 12150. |
| IIOP_CICS_SERVER_PORT | Default value is 12151. |

# Customizing the CICS Transformer Service for Server Mode

**Overview**

This subsection is only relevant if you want to use the CICS transformer service in server mode with Artix Mainframe. It describes the customization tasks to be performed before using the CICS transformer in non-secure server mode.

**Note:** If you need to perform the tasks in this subsection, perform them in the order in which they are presented. Before you proceed, make sure the tasks in "Standard Customization Tasks" on page 33 have already been completed.

**Step 1: Avoid known problems**

IONA recommends that the PTFs listed in "Installation Prerequisites" on page 15 are applied, to avoid known problems.

**Step 2: Configure EXCI or APPC for CICS**

To use the CICS transformer service, either of the following must be enabled for CICS:

- EXCI
- APPC

For details of how to configure EXCI for CICS see the IBM publication: *CICS External Interfaces Guide, SC34-6449*.

For details of how to configure APPC for CICS, see the IBM publication: *MVS Planning: APPC/MVS Management, SA22-7599*. Additionally, for specific details on the use of APPC by CICS, see the chapter on defining APPC links in the IBM publication: *CICS Intercommunication Guide, SC33-1695*.

**Note:** When using EXCI, your CICS programs use the default transaction, "`ARX1`". Please make sure that users are provided with the necessary authorization to run this transaction.

**Step 3: Configure the CICS 3270 bridge facility**

The Link3270 mechanism must be enabled if you want to expose existing BMS-based CICS applications as Web services or CORBA objects. The CICS transformer service uses the Link3270 component of the CICS 3270 bridge facility to communicate with CICS when forwarding Web service or CORBA client requests to existing BMS-based CICS applications. For details on how to enable the Link3270 mechanism, see the section on Bridging to 3270 transactions in the IBM publication *CICS External Interfaces Guide, SC34-6449.*

### Step 4: Define required resources to CICS

Before you can run the CICS transformer service in server mode, you must perform a number of additional steps to enable CICS to support it. Depending on your installation, one or all of these tasks might already have been completed. You must verify this with the systems programmer responsible for CICS at your site. See the Artix Mainframe *Administrator's Guide* for more information on these tasks:

- Check whether the latest CICS Language Environment (LE) support is installed in your CICS region. See the IBM publication: *Language Environment for z/OS Customization* for details on installing LE support in CICS.
- Check whether support for the C++ standard classes is explicitly defined to CICS. See the IBM publication *z/OS C/C++ Programming Guide* for details of the steps required to run C++ application programs under CICS.

A sample job is provided in `artixhlq.JCLLIB(ARTIXCSD)` to run DFHCSDUP (which is the CICS offline resource definition utility) to define the CICS resources used by the sample jobs and demonstrations. You can run this job, or just use it as a reference when defining the resources online with the CEDA transaction.

When the resources have been defined, use CEDA to install the whole group. If you decide to run the job, first change the JCL to reflect the proper CICS high-level qualifier in use at your site.

### Step 5: Customize CICS JCL

Follow these steps to customize the CICS JCL:

- Check whether the `CEE.SCEERUN` and `CBC.SCLBDLL` libraries are already in the STEPLIB concatenation for the CICS region. If not, add them as follows:

```
DD DSN=CEE.SCEERUN,DISP=SHR
DD DSN=CBC.SCLBDLL,DISP=SHR
```

- Check whether the `CEE.SCEERUN` library is already in the DFHRPL for the CICS region. If not, add it as follows:

  ```
  DD DSN=CEE.SCEERUN,DISP=SHR
  ```

- Check whether `CEEMSG` and `CEEOUT` entries are already defined in the JCL for the CICS region. If not, add them as follows, to make sure you receive all output from your CICS servers:

  ```
  CEEMSG DD SYSOUT=*
  CEEOUT DD SYSOUT=*
  ```

You must recycle CICS to pick up these changes.

**Note:** Artix uses the default location for the `CEE.SCEERUN` and `CBC.SCLBDLL` system libraries. You might need to update these locations for your system. See Table 2 on page 33 for more details.

### Step 6: CICS Security

The CICS transformer service in server mode uses standard CICS security mechanisms to communicate with the CICS regions. See the Artix Mainframe *Administrator's Guide* for a detailed description of security considerations involved in using the CICS transformer service, and a review of general Artix and CICS security implications.

### Step 7: Verify transformer configuration prerequisites

Verify that the configuration variables in the `cicsa` scope of your configuration file have been changed to match those specified in the CICS control region that you are connecting to. In particular, make sure you have specified the location of the transformer mapping member to be used. For details on how to do this, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide*.

### Step 8: Start the CICS transformer service in server mode

You are now ready to start the CICS transformer service in server mode. Follow these steps:

1. Edit the JCL in `*artixhlq*.JCLLIB(CICST)`, to change the default high-level qualifier so that it reflects the proper value for your installation.
2. Depending on which transports you want the transformer service to support, do one of the following:

- If you want to use HTTP, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM=''
```

- If you want to use WebSphere MQ, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM='-ORBname iona_services.cicsa.use_mq'
```

- If you want to use IIOP, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM='-ORBname iona_services.cicsa.use_iiop'
```

3. Submit the *artixhlq*.JCLLIB(CICST) job to start the CICS transformer service in server mode.

### Step 9: Verify the transformer service is running

Enter the following in a web browser from any client computer to confirm that the CICS transformer service has started successfully and is accepting requests:

```
http://remotehost:port/ionasoap/
```

In the preceding example, remotehost represents the z/OS TCP/IP hostname, and port represents the port on which the transformer service is listening for client requests. You should see a list of the registered clients available to you in the transformer service. Select any service and you should see the generated WSDL for that service in your browser window.

### Step 10: Run the supplied demonstration

Run the supplied demonstration to ensure that all installation and configuration has been completed successfully. See the Artix Mainframe *Getting Started* guide for details on how to run the supplied demonstration for the CICS transformer service.

# Customizing the CICS Transformer Service for Client Mode

**Overview**

This subsection is only relevant if you want to use the CICS transformer service in client mode with Artix Mainframe. It describes the customization tasks to be performed before using the CICS transformer in non-secure client mode.

**Note:** If you need to perform the tasks in this subsection, perform them in the order in which they are presented. Before you proceed, make sure the tasks in "Standard Customization Tasks" on page 33 have already been completed.

**Step 1: Avoid known problems**

IONA recommends that the PTFs listed in "Installation Prerequisites" on page 15 are applied, to avoid known problems.

**Step 2: Configure APPC or cross memory communication for CICS**

To use the CICS transformer service in client mode with APPC, APPC communication must be enabled for CICS. For details of how to configure APPC for CICS, see the IBM publication: *MVS Planning: APPC/MVS Management, SA22-7599*.

Additionally, for specific details on the use of APPC by CICS, see the chapter on defining APPC links in the IBM publication: *CICS Intercommunication Guide, SC33-1695*.

To use the CICS transformer service in client mode with cross memory communication, the transformer service must be APF-authorized, and the transformer service must run in a non-swappable address space. See the Artix Mainframe *Administrator*'s *Guide* for details on performing these tasks.

**Step 3: Define transformer service APPC/MVS side information**

To use the transformer service in client mode with APPC, you must define a symbolic destination name in the APPC/MVS Side Information data set. Although JCL is not provided to do this in your product installation, the Artix Mainframe *Administrator's Guide* provides an example of how to do this using a symbolic destination name of ARXCLNT1.

### Step 4A: Verify transformer service configuration with APPC

Follow these steps to verify transformer service configuration with APPC:

1. Verify that the configuration variables in the `cics_client` scope of your configuration member are valid for your installation. In particular, verify that the following configuration variable matches the transformer service APPC/MVS Side Information DESTNAME you specified in Step 3. For example:

   ```
   plugins:amtp_appc:symbolic_destination = "ARXCLNT1";
   ```

   For details of how to change configuration, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide*.

2. Review the following client configuration parameters shipped in *`artixhlq`*`.JCLLIB(MFACLINK)` and make any changes that are required for your site:

| Parameter | Description |
|---|---|
| LOGLVL | The value specified determines the level of event logging that is enabled. Valid values are:<br>`0`—`LOG_NONE` (no logging is performed).<br>`1`—`LOG_ERROR` (only log errors).<br>`2`—`LOG_WARNING` (log warnings and errors).<br>`3`—`LOG_INFO_HIGH` (log high priority informational messages, warnings and errors).<br>`4`—`LOG_INFO_MED` (log medium and high priority informational messages, warnings and errors).<br>`5`—`LOG_INFO_LOW` (log low, medium and high priority informational informational messages, warnings and errors)<br>`6`—`LOG_INFO_ALL` (log all messages). |

| Parameter | Description |
|---|---|
| MAXSEG | The Artix runtime in CICS builds up APPC segments of this size. For APPC, multiple segments of this size are used to transmit data. The specified value must be a multiple of 8. The minimum allowed value is 32 bytes. The maximum allowed value is 32760. The default is 32760. |
| TIMEOUT | This value is not used by the Artix runtime in CICS. Leave it set to the default value as shipped. |
| SYMBDST | The value specified must match the value in the transformer service APPC/MVS Side Information DESTNAME you specified in Step 3. |
| LOCALLU | This value is not used by the Artix runtime in CICS. Leave it set to the default value as shipped. |

If you need to change any of the shipped values, you must assemble and relink the new configuration into *artixhlq*.MFA.LOADLIB (ORXWCFG1). Edit the JCL in *artixhlq*.JCLLIB(MFACLINK) to change the default high-level qualifier, so that it reflects the proper value for your installation, and then submit the JCL.

### Step 4B: Verify transformer service configuration with cross memory communication

Follow these steps to verify transformer service configuration with cross memory communication:

1. Verify that the configuration variables in the cics_client.cross_memory scope of your configuration member are valid for your installation. In particular, verify that the following configuration variable matches the SYMBDST client configuration parameter defined in *artixhlq*.JCLLIB(MFACLINK). For example:

```
plugins:amtp_xmem:symbolic_destination = "ARXCLNT1";
```

For details of how to change configuration, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide*.

2. Review the following client configuration parameters shipped in `*artixhlq*.JCLLIB(MFACLINK)` and make any changes that are required for your site:

| Parameter | Description |
|---|---|
| LOGLVL | The value specified determines the level of event logging that is enabled. Valid values are:<br>`0`—`LOG_NONE` (no logging is performed).<br>`1`—`LOG_ERROR` (only log errors).<br>`2`—`LOG_WARNING` (log warnings and errors).<br>`3`—`LOG_INFO_HIGH` (log high priority informational messages, warnings and errors).<br>`4`—`LOG_INFO_MED` (log medium and high priority informational messages, warnings and errors).<br>`5`—`LOG_INFO_LOW` (log low, medium and high priority informational informational messages, warnings and errors)<br>`6`—`LOG_INFO_ALL` (log all messages). |
| MAXSEG | The Artix runtime in CICS builds up buffers of this size. If the data being transported is greater than this size, multiple buffers of this size are used to transmit data. The specified value must be a multiple of 8. The minimum allowed value is `64` bytes. The maximum allowed value is `32760`. The default is `32760`. |

| Parameter | Description |
|---|---|
| TIMEOUT | The Artix runtime in CICS uses the PROGRAM CALL (PC) assembler instruction to invoke a PC routine to move data between CICS and the transformer service running in client mode. Three PC calls are made when processing a client invocation (that is, Send data, Receive reply buffer count, and Receive reply). The timeout value governs how long it takes to make these three PC calls. If the three calls cannot be made within the configured timeout value, a `COMM_FAILURE` exception is raised. The timeout value is specified in seconds. (This differs from APPC where the timout is specified in minutes.) The default is `5` seconds. |
| SYMBDST | The value specified must match the value in the transformer service configuration item `plugins:amtp_xmem:symbolic_destination`. |
| LOCALLU | This value must be set to “IT_XMEM”. This setting causes the Artix runtime in CICS to use cross memory communication for interacting with the transformer service running in client mode. |

If you need to change any of the shipped values, you must assemble and relink the new configuration into `*artixhlq*.MFA.LOADLIB(ORXWCFG1)`. Edit the JCL in `*artixhlq*.JCLLIB(MFACLINK)` to change the default high-level qualifier, so that it reflects the proper value for your installation, and then submit the JCL.

**Step 5: Define required resources to CICS**

Before you can run the CICS transformer service in client mode, you must perform a number of additional steps to enable CICS to support it. Depending on your installation, one or all of these tasks might already have been completed. You must verify this with the systems programmer responsible for CICS at your site. See the Artix Mainframe *Administrator's Guide* for more details on these tasks:

- Check whether the latest CICS Language Environment (LE) support is installed in your CICS region. See the IBM publication: *Language Environment for z/OS Customization* for details on installing LE support in CICS.
- Check whether support for the C++ standard classes is explicitly defined to CICS. See the IBM publication *z/OS C/C++ Programming Guide* for details of the steps required to run C++ application programs under CICS.

A sample job is provided in *artixhlq*.JCLLIB(ARTIXCSD) to run DFHCSDUP (which is the CICS offline resource definition utility) to define the CICS resources used by the sample jobs and demonstrations. You can run this job, or just use it as a reference when defining the resources online with the CEDA transaction.

When the resources have been defined, use CEDA to install the whole group. If you decide to run the job, first change the JCL to reflect the proper CICS high-level qualifier in use at your site.

## Step 6: Customize CICS JCL

To use the transformer service in client mode with CICS:

- Add the following libraries to the CICS region's DFHRPL, as follows:

```
DD DSN=artixhlq.MFA.LOADLIB.DISP=SHR
DD DSN=artixhlq.DEMOS.CICS.PLI.LOADLIB,DISP=SHR
DD DSN=artixhlq.DEMOS.CICS.CBL.LOADLIB,DISP=SHR
```

- Check whether the CEE.SCEERUN and CBC.SCLBDLL libraries are already in the STEPLIB concatenation for the CICS region. If not, add them as follows:

```
DD DSN=CEE.SCEERUN,DISP=SHR
DD DSN=CBC.SCLBDLL,DISP=SHR
```

- Check whether the CEE.SCEERUN library is already in the DFHRPL for the CICS region. If not, add it as follows:

```
DD DSN=CEE.SCEERUN,DISP=SHR
```

- Check whether CEEMSG and CEEOUT entries are already defined in the JCL for the CICS region. If not, add them as follows to make sure you receive all output from your CICS servers:

```
CEEMSG DD SYSOUT=*
CEEOUT DD SYSOUT=*
```

You must recycle CICS to pick up these changes.

### Step 7: CICS Security

The CICS transformer service in client mode uses standard CICS security mechanisms to communicate with the CICS regions. See the Artix Mainframe *Administrator's Guide* for a detailed description of security considerations involved in using the CICS transformer service, and a review of general Artix and CICS security implications.

### Step 8: Start the transformer service in client mode

You are now ready to start the transformer service in client mode.

1. Edit the JCL in *artixhlq*.JCLLIB(CICSC), to change the default high-level qualifier, so that it reflects the proper value for your installation.
2. Depending on which transports you want the transformer service to support, do one of the following:
   - If you want to use APPC on the client side with HTTP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

     ```
     PPARM='-ORBname iona_services.cics_client'
     ```

   - If you want to use APPC on the client side with WebSphere MQ on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

     ```
     PPARM='-ORBname iona_services.cics_client.use_mq'
     ```

   - If you want to use cross memory communication on the client side with HTTP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

     ```
     PPARM='-ORBname iona_services.cics_client.cross_memory'
     ```

- If you want to use cross memory communication on the client side with WebSphere MQ on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM='-ORBname iona_services.cics_client.cross_memory.use_mq'
```

- If you want to use IIOP on the server side, update the PPARM JCL to point to the iiop configuration scope:

```
PPARM='-ORBname iona_services.cics_client.use_iiop'
```

3. Submit this JCL to start the transformer service in client mode.

**Step 9: Verify the transformer service is running**

Enter the following in a web browser from any client computer to confirm that the CICS transformer service has started successfully and is accepting requests:

```
http://remotehost:port/
```

In the preceding example, *remotehost* represents the z/OS TCP/IP hostname, and *port* represents the port on which the transformer service is listening for deployments from Artix Designer.

**Step 10: Run the supplied demonstration**

Run the supplied demonstration to make sure all installation and configuration has been completed successfully. See the Artix Mainframe *Getting Started* guide for details on how to run the supplied demonstration for the CICS transformer service.

# Customizing the IMS Transformer Service for Server Mode

**Overview**

This subsection is only relevant if you want to use the IMS transformer service in server mode with Artix Mainframe. It describes the customization tasks to be performed before using the IMS transformer in non-secure server mode.

> **Note:** If you need to perform the tasks in this subsection, perform them in the order in which they are presented. Before you proceed, make sure the tasks in "Standard Customization Tasks" on page 33 have already been completed.

**Step 1: Avoid known problems**

IONA recommends that the PTFs listed in "Installation Prerequisites" on page 15 are applied, to avoid known problems.

**Step 2: Configure OTMA or APPC for IMS**

To use the IMS transformer service, either of the following must be enabled for IMS:

- OTMA and the OTMA Callable Interface
- APPC

For details on how to configure OTMA for IMS, see the IBM publication *Open Transaction Manager Access Guide and Reference, SC18-7829*.

For details of how to configure APPC for IMS, see the IBM publication *MVS Planning: APPC/MVS Management, SA22-7599*. Additionally, for specific details on the use of APPC by IMS, see the chapter on administration of APPC/IMS and LU 6.2 devices in the IBM publication *IMS Administration Guide: Transaction Manager, SC18-7808*.

**Step 3: Verify transformer configuration prerequisites**

Verify that the configuration variables in the `imsa` scope of your configuration file have been changed to match those specified in the IMS control region that you are connecting to. In particular, make sure you have specified the location of the transformer mapping member to be used. For details on how to do this, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide*.

### Step 4: Customize IMS JCL

Check whether the following entries are already defined in the IMS message region's JCL. If not, add them to make sure you receive all output from your IMS servers. Recycle the message regions to pick up these libraries:

```
SYSPRINT DD SYSOUT=*
CEEDUMP DD SYSOUT=*
CEEOUT DD SYSOUT=*
SYSOUT DD SYSOUT=*
```

### Step 5: Start the IMS transformer service in server mode

You are now ready to start the IMS transformer service in server mode. Follow these steps:

1. Edit the JCL in `*artixhlq*.JCLLIB(IMST)` to change the default high-level qualifier so that it reflects the proper value for your installation.
2. Depending on which transports you want the transformer service to support, do one of the following:
   - If you want to use HTTP, ensure that the PPARM JCL symbolic points to the following configuration scope:

   ```
   PPARM=''
   ```

   - If you want to use WebSphere MQ, ensure that the PPARM JCL symbolic points to the following configuration scope:

   ```
   PPARM='-ORBname iona_services.imsa.use_mq'
   ```

   - If you want to use IIOP, ensure that the PPARM JCL symbolic points to the following configuration scope:

   ```
   PPARM='-ORBname iona_services.imsa.use_iiop'
   ```

3. Submit the `HLQ.ARTIX51.JCLLIB(IMST)` job to start the IMS transformer service.

**Step 6: Verify the transformer service is running**

Enter the following in a web browser from any client computer to confirm that the IMS transformer service has started successfully and is accepting requests:

```
http://remotehost:port/ionasoap/
```

In the preceding example, *remotehost* represents the z/OS TCP/IP hostname, and *port* represents the port on which the transformer service is listening for client requests. You should see a list of the registered clients available to you in the transformer service. Select any service and you should see the generated WSDL for that service in your browser window.

**Step 7: Run the supplied demonstration**

Run the supplied demonstration to make sure all installation and configuration has been completed successfully. See the Artix Mainframe *Getting Started* guide for details on how to run the supplied demonstration for the IMS transformer service.

# Customizing the IMS Transformer Service for Client Mode

**Overview**

This subsection is only relevant if you want to use the IMS transformer service in client mode with Artix Mainframe. It describes the customization tasks to be performed before using the transformer service in non-secure client mode.

**Step 1: Avoid known problems**

IONA recommends that the PTFs listed in "Installation Prerequisites" on page 15 are applied, to avoid known problems.

**Step 2: Configure APPC or cross memory communication for IMS**

To use the transformer service in client mode with APPC, APPC communication must be enabled for IMS. For details of how to configure APPC for IMS see the IBM publication *MVS Planning: APPC/MVS Management, SA22-7599*.

Additionally, for specific details on the use of APPC by IMS, see the chapter on administration of APPC/IMS and LU 6.2 devices in the IBM publication *IMS Administration Guide: Transaction Manager, SC18-7808*.

To use the IMS transformer service in client mode with cross memory communication, the transformer service must be APF-authorized, and the transformer service must run in a non-swappable address space. See the Artix Mainframe *Administrator's Guide* for the details on performing these tasks.

**Step 3: Define transformer service APPC/MVS side information**

To use the transformer service with APPC in client mode, you must define a symbolic destination name in the APPC/MVS Side Information data set. Although JCL is not provided to do this in your product installation, the Artix Mainframe *Administrator's Guide* provides an example of doing this using a symbolic destination name of ARXCLNT1.

### Step 4A: Verify transformer service configuration

Follow these steps to verify transformer service configuration with APPC:

1. Verify that the configuration variables in the `ims_client` scope of your configuration member are valid for your installation. In particular, verify that the following configuration variable matches the transformer service APPC/MVS Side Information DESTNAME you specified in Step 3. For example:

   ```
   plugins:amtp_appc:symbolic_destination = "ARXCLNT1";
   ```

   For details on how to change configuration, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide.*

2. Review the following client configuration parameters shipped in *`artixhlq`*`.JCLLIB(MFACLINK)` and make any changes that are required:

| Parameter | Description |
|---|---|
| LOGLVL | The value specified determines the level of event logging that is enabled. Valid values are:<br>`0—LOG_NONE` (no logging is performed).<br>`1—LOG_ERROR` (only log errors).<br>`2—LOG_WARNING` (log warnings and errors).<br>`3—LOG_INFO_HIGH` (log high priority informational messages, warnings and errors).<br>`4—LOG_INFO_MED` (log medium and high priority informational messages, warnings and errors).<br>`5—LOG_INFO_LOW` (log low, medium and high priority informational informational messages, warnings and errors)<br>`6—LOG_INFO_ALL` (log all messages). |

| Parameter | Description |
|---|---|
| MAXSEG | The Artix runtime in IMS builds up APPC segments of this size. For APPC, multiple segments of this size are used to transmit data. The specified value must be a multiple of 8. The minimum allowed value is 32 bytes. The maximum allowed value is 32760. The default is 32760. |
| TIMEOUT | The value specified determines the length of time (in minutes) that the Artix runtime in IMS allows an APPC receive call to wait, to receive data from the transformer service, before it is timed out. The specified value must be in the range 0–1440. A value of 0 means no timeout. The default is 5 minutes. |
| SYMBDST | The value specified must match the value in the transformer service APPC/MVS Side Information DESTNAME you specified in Step 3. |
| LOCALLU | The value specified must match the transformer service IMS LU name. This is the LU name used for APPC communications in IMS. |

If you need to change any of the shipped values, you must assemble and relink the new configuration into *artixhlq*.MFA.LOADLIB(ORXWCFG1). Edit the JCL in *artixhlq*.JCLLIB(MFACLINK) to change the default high-level qualifier so that it reflects the proper value for your installation and then submit the JCL.

### Step 4B: Verify transformer service configuration with cross memory communication

Follow these steps to verify transformer service configuration with cross memory communication:

1. Verify that the configuration variables in the `ims_client.cross_memory` scope of your configuration member are valid for your installation. In particular, verify that the following configuration variable matches the SYMBDST client configuration parameter defined in *artixhlq*.JCLLIB(MFACLINK). For example:

   ```
   plugins:amtp_xmem:symbolic_destination = "ARXCLNT1";
   ```

   For details of how to change configuration, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide*.

2. Review the following client configuration parameters shipped in *artixhlq*.JCLLIB(MFACLINK) and make any changes that are required for your site:

| Parameter | Description |
|---|---|
| LOGLVL | The value specified determines the level of event logging that is enabled. Valid values are:<br>0—LOG_NONE (no logging is performed).<br>1—LOG_ERROR (only log errors).<br>2—LOG_WARNING (log warnings and errors).<br>3—LOG_INFO_HIGH (log high priority informational messages, warnings and errors).<br>4—LOG_INFO_MED (log medium and high priority informational messages, warnings and errors).<br>5—LOG_INFO_LOW (log low, medium and high priority informational informational messages, warnings and errors)<br>6—LOG_INFO_ALL (log all messages). |

| Parameter | Description |
|---|---|
| MAXSEG | The Artix runtime in IMS builds up buffers of this size. If the data being transported is greater than this size, multiple buffers of this size are used to transmit data. The specified value must be a multiple of 8. The minimum allowed value is `64` bytes. The maximum allowed value is `32760`. The default is `32760`. |
| TIMEOUT | The Artix runtime in IMS uses the PROGRAM CALL (PC) assembler instruction to invoke a PC routine to move data between IMS and the transformer service running in client mode. Three PC calls are made when processing a client invocation (that is, Send data, Receive reply buffer count, and Receive reply). The timeout value governs how long it takes to make these three PC calls. If the three calls cannot be made within the configured timeout value, a `COMM_FAILURE` exception is raised. The timeout value is specified in seconds. (This differs from APPC where the timout is specified in minutes.) The default is `5` seconds. |
| SYMBDST | The value specified must match the value in the transformer service configuration item `plugins:amtp_xmem:symbolic_destination`. |
| LOCALLU | This value must be set to "IT_XMEM". This setting causes the Artix runtime in IMS to use cross memory communication for interacting with the transformer service running in client mode. |

If you need to change any of the shipped values, you must assemble and relink the new configuration into *artixhlq*.MFA.LOADLIB(ORXWCFG1). Edit the JCL in *artixhlq*.JCLLIB(MFACLINK) to change the default high-level qualifier so that it reflects the proper value for your installation and then submit the JCL.

### Step 5: Customize IMS JCL

To use the transformer service in client mode with IMS:

- Add the following libraries to the IMS region's STEPLIB concatenation, as follows:

```
DD DSN=artixhlq.MFA.LOADLIB.DISP=SHR
DD DSN=artixhlq.DEMOS.IMS.PLI.LOADLIB,DISP=SHR
DD DSN=artixhlq.DEMOS.IMS.CBL.LOADLIB,DISP=SHR
```

- Check whether the following entries are already defined in the IMS message region's JCL. If not, add them to make sure you receive all output from your IMS clients. Recycle the message regions to pick up these libraries:

```
SYSPRINT DD SYSOUT=*
CEEDUMP DD SYSOUT=*
CEEOUT DD SYSOUT=*
SYSOUT DD SYSOUT=*
```

- Check whether the `CEE.SCEERUN` library is already in the STEPLIB concatenation for the IMS region. If not, add it as follows:

```
DD DSN=CEE.SCEERUN,DISP=SHR
```

### Step 6: Start the transformer service in client mode

You are now ready to start the transformer service in client mode.

1. Edit the JCL in *artixhlq*`.JCLLIB(IMSC)`, to change the default high-level qualifier so that it reflects the proper value for your installation.
2. Depending on which transports you want the transformer service to support, do one of the following:
   - If you want to use APPC on the client side with HTTP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

```
PPARM='-ORBname iona_services.ims_client'
```

- If you want to use APPC on the client side with WebSphere MQ on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

```
PPARM='-ORBname iona_services.ims_client.use_mq'
```

- If you want to use cross memory communication on the client side with HTTP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

```
PPARM='-ORBname iona_services.ims_client.cross_memory'
```

- If you want to use cross memory communication on the client side with WebSphere MQ on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM='-ORBname iona_services.ims_client.cross_memory.use_mq'
```

- If you want to use IIOP on the server side, update the PPARM JCL to point to the iiop configuration scope:

```
PPARM='-ORBname iona_services.ims_client.use_iiop'
```

3. Submit this JCL to start the transformer service in client mode.

### Step 7: Verify the transformer service is running

Enter the following in a web browser from any client computer to confirm that the IMS transformer service has started successfully and is accepting requests:

```
http://remotehost:port/
```

In the preceding example, *remotehost* represents the z/OS TCP/IP hostname, and port represents the port on which the transformer service is listening for deployments from Artix Designer.

### Step 8: Run the supplied demonstration

Run the supplied demonstration to make sure all installation and configuration has been completed successfully. See the Artix Mainframe *Getting Started* guide for details on how to run the supplied demonstrations for the IMS transformer service.

## Customizing the Batch Transformer Service for Client Mode

### Overview

This subsection is only relevant if you want to use the batch transformer service in client mode with Artix Mainframe. It describes the customization tasks to be performed before using the batch transformer service in client mode.

### Step 1: Avoid known problems

To avoid known problems, IONA recommends that you apply the PTFs listed in "Installation Prerequisites" on page 15.

### Step 2: Configure APPC or cross memory communication for batch

To use the batch transformer service in client mode with APPC, APPC communication must be enabled. For details on configuring MVS for APPC, see the IBM publication *MVS Planning: APPC/MVS Management, SA22-7599*.

To use the batch transformer service in client mode with cross memory communication, the transformer service must be APF-authorized, and the transformer service must run in a non-swappable address space. See the Artix Mainframe *Administrator's Guide* for details on performing these tasks.

### Step 3: Define transformer service APPC/MVS side information

To use the batch transformer service in client mode with APPC, you must define a symbolic destination name in the APPC/MVS Side Information data set.

Although JCL is not provided to do this in your product installation, the Artix Mainframe *Administrator's Guide* provides an example of how to do this using a symbolic destination name of `ARXCLNT1`.

### Step 4A: Verify transformer service configuration

Follow these steps to verify transformer service configuration with APPC:

1. Verify that the configuration variables in the `batch_client` scope of your configuration member are valid for your installation. In particular, verify that the following configuration variable matches the transformer service APPC/MVS Side Information DESTNAME you specified in Step 3. For example:

```
plugins:amtp_appc:symbolic_destination = "ARXCLNT1";
```

For details on changing configuration, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide*.

2. Review the following client configuration parameters shipped in *artixhlq*.JCLLIB(MFACLINK) and make any changes required for your site:

| Parameter | Description |
|---|---|
| LOGLVL | The value specified determines the level of event logging that is enabled. Valid values are:<br>0—LOG_NONE (no logging is performed).<br>1—LOG_ERROR (only log errors).<br>2—LOG_WARNING (log warnings and errors).<br>3—LOG_INFO_HIGH (log high priority informational messages, warnings and errors).<br>4—LOG_INFO_MED (log medium and high priority informational messages, warnings and errors).<br>5—LOG_INFO_LOW (log low, medium and high priority informational informational messages, warnings and errors)<br>6—LOG_INFO_ALL (log all messages). |
| MAXSEG | The Artix runtime in batch builds up APPC segments of this size. For APPC, multiple segments of this size are used to transmit data. The specified value must be a multiple of 8. The minimum allowed value is 32 bytes. The maximum allowed value is 32760. The default is 32760. |
| TIMEOUT | The value specified determines the length of time (in minutes) that the Artix runtime in batch allows an APPC receive call to wait, to receive data from the transformer service, before it is timed out. The specified value must be in the range 0–1440. A value of 0 means no timeout. The default is 5 minutes. |

| Parameter | Description |
|---|---|
| SYMBDST | The value specified must match the value in the transformer service APPC/MVS Side Information. |
| LOCALLU | The value specified must match the transformer service batch client LU name. This is the LU name used for APPC communications in MVSLU01. |

If you need to change any of the shipped values, you must assemble and relink the new configuration into *artixhlq*..LOADLIB(ORXWCFG1). Edit the JCL in *artixhlq*.JCLLIB(MFACLINK) to change the default high-level qualifier so that it reflects the proper value for your installation, and then submit the JCL.

**Note:** This job is also used to update the configuration details used in IMS or CICS. Please update the CFGPDS variable at the top of the job by commenting out the first value and uncommenting the second:

```
//*      SET CFGPDS=&ARTIX..MFA.LOADLIB
//       SET CFGPDS=&ARTIX..LOADLIB
```

### Step 4B: Verify transformer service configuration with cross memory communication

Follow these steps to verify transformer service configuration with cross memory communication:

1. Verify that the configuration variables in the ims_client.cross_memory scope of your configuration member are valid for your installation. In particular, verify that the following configuration variable matches the SYMBDST client configuration parameter defined in *artixhlq*.JCLLIB(MFACLINK). For example:

```
plugins:amtp_xmem:symbolic_destination = "ARXCLNT1";
```

For details on changing configuration, and the defaults used when the entries are not specified via configuration, see the Artix Mainframe *Administrator's Guide*.

2. Review the following client configuration parameters shipped in *artixhlq*.JCLLIB(MFACLINK) and make any changes that are required for your site:

| Parameter | Description |
|---|---|
| LOGLVL | The value specified determines the level of event logging that is enabled. Valid values are:<br>0—LOG_NONE (no logging is performed).<br>1—LOG_ERROR (only log errors).<br>2—LOG_WARNING (log warnings and errors).<br>3—LOG_INFO_HIGH (log high priority informational messages, warnings and errors).<br>4—LOG_INFO_MED (log medium and high priority informational messages, warnings and errors).<br>5—LOG_INFO_LOW (log low, medium and high priority informational informational messages, warnings and errors)<br>6—LOG_INFO_ALL (log all messages). |
| MAXSEG | The Artix runtime in batch builds up buffers of this size. If the data being transported is greater than this size, multiple buffers of this size are used to transmit data. The specified value must be a multiple of 8. The minimum allowed value is 64 bytes. The maximum allowed value is 32760. The default is 32760. |

| Parameter | Description |
|---|---|
| TIMEOUT | The Artix runtime in batch uses the PROGRAM CALL (PC) assembler instruction to invoke a PC routine to move data between the batch client and the transformer service running in client mode. Three PC calls are made when processing a client invocation (that is, Send data, Receive reply buffer count, and Receive reply). The timeout value governs how long it takes to make these three PC calls. If the three calls cannot be made within the configured timeout value, a COMM_FAILURE exception is raised. The timeout value is specified in seconds. (This differs from APPC where the timout is specified in minutes.) The default is 5 seconds. |
| SYMBDST | The value specified must match the value in the transformer service configuration item<br>plugins:amtp_xmem:symbolic_destination |
| LOCALLU | This value must be set to “IT_XMEM”. This setting causes the Artix runtime to use cross memory communication for interacting with the transformer service running in client mode. |

If you need to change any of the shipped values, you must assemble and relink the new configuration into *artixhlq*.LOADLIB(ORXWCFG1). Edit the JCL in *artixhlq*.JCLLIB(MFACLINK) to change the default high-level qualifier so that it reflects the proper value for your installation, and then submit the JCL.

**Note:** This job is also used to update the configuration details used in IMS or CICS. Please update the CFGPDS variable at the top of the job by commenting out the first value and uncommenting the second:

```
//*          SET CFGPDS=&ARTIX..MFA.LOADLIB
//           SET CFGPDS=&ARTIX..LOADLIB
```

### Step 5: Start the batch transformer service in client mode

You are now ready to start the batch transformer service in client mode.

1. Edit the JCL in *artixhlq*.JCLLIB(BATCHC), to change the default high-level qualifier so that it reflects the proper value for your installation.
2. Depending on which transports you want the transformer service to support, do one of the following:
    - If you want to use APPC on the client side with HTTP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

```
PPARM='-ORBname iona_services.batch_client'
```

    - If you want to use APPC on the client side with WebSphere MQ on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

```
PPARM='-ORBname iona_services.batch_client.use_mq'
```

    - If you want to use APPC on the client side with IIOP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM='-ORBname iona_services.batch_client.use_iiop'
```

    - If you want to use cross memory communication on the client side with HTTP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope::

```
PPARM='-ORBname iona_services.batch_client.cross_memory'
```

    - If you want to use cross memory communication on the client side with WebSphere MQ on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM='-ORBname iona_services.batch_client.cross_memory.use_mq'
```

- If you want to use cross memory communication on the client side with IIOP on the server side, ensure that the PPARM JCL symbolic points to the following configuration scope:

```
PPARM='-ORBname iona_services.batch_client.cross_memory.use_iiop'
```

3. Submit this JCL to start the transformer service in client mode.

**Step 6: Verify the transformer service is running**

Enter the following in a web browser from any client computer to confirm that the batch transformer service has started successfully and is accepting requests:

```
http://remotehost:port/
```

In the preceding example, *remotehost* represents the z/OS TCP/IP hostname, and *port* represents the port on which the transformer service is listening for deployments from Artix Designer.

**Step 7: Run the supplied demonstration**

Run the supplied demonstration to make sure all installation and configuration has been completed successfully. See the Artix Mainframe *Getting Started* guide for details on running the supplied demonstrations for the batch transformer service in client mode.

## Customizing the DB2 Gateway Service

### Overview

This subsection is only relevant if you want to use the Artix Mainframe DB2 Gateway service. It describes the customization tasks to be performed before using the Artix Mainframe DB2 Gateway service.

### Step 1: Verify that ODBC support has been installed

If you are not sure whether ODBC support has been installed and enabled for your DB2 system, use the DB2 sample application in `*db2hlq*.SDSNSAMP(DSNTEJ8)` to verify that ODBC/CLI is working correctly (where *`db2hlq`* represents your DB2 high-level qualifier).

**Note:** The DB2 gateway service cannot function without ODBC/CLI.

If ODBC is not working, follow the steps to configure ODBC as described in the IBM publication: *DB2 Universal Database for z/OS: ODBC Guide and Reference*.

### Step 2: Verify DB2 Gateway service configuration

Follow these steps to verify the DB2 Gateway service configuration:

1. Verify that the configuration variables in the DB2 scope of your configuration member are valid for your installation. In particular, verify that the following configuration variable matches the name of the database to which the Artix DB2 Gateway should connect:

   ```
   configuration:database_name="SAMPLE";
   ```

2. Ensure that a suitable TCP/IP port has been specified for the DB2 gateway at the start of the configuration file.

### Step 3: Start the DB2 Gateway service

You are now ready to start the Artix Mainframe DB2 gateway service:

1. Edit the JCL in *`artixhlq`*`.JCLLIB(DB2GW)` to change the default high-level qualifier so that it reflects the proper value for your installation.
2. Submit this JCL to start the DB2 Gateway service.

**Step 4: Verify the DB2 Gateway service is running**

Enter the following in a web browser from any client computer to confirm that the DB2 Gateway service has started successfully and is accepting requests:

```
http://remotehost:port/ionasoap/
```

In the preceding example, *remotehost* represents the z/OS TCP/IP hostname, and *port* represents the port on which the DB2 gateway service is listening for deployments from Artix Designer.

**Step 5: Run the supplied demonstration**

Run the supplied demonstration to make sure all installation and configuration has been completed successfully. See the Artix Mainframe *Getting Started* guide for details on running the supplied demonstrations.

# SSL Customization Tasks

## Overview

This subsection is only relevant if you want to enable the IMS or CICS transformer service to use secure sockets layer (SSL) security. It describes the customization tasks to be performed before you can use the IMS or CICS transformer service in secure mode.

> **Note:** If you need to perform the tasks in this subsection, perform them in the order in which they are presented. Before you proceed, make sure that the tasks in "Customizing the CICS Transformer Service for Server Mode" on page 42 have already been completed.

## Step 1: Generate SSL certificates

To generate SSL certificates:

1. Edit the JCL in *artixhlq*.JCLLIB(GENCERT) to change the user ID and high-level qualifier to reflect the proper values for your installation.
2. Submit *artixhlq*.JCLLIB(GENCERT) to generate the SSL certificates.

> **Note:** If you have existing SSL certificates under your user ID, it is recommended that you either delete them before running the GENCERT JCL, or at least make sure the generated certificates have a different name (DN) and label than the existing certificates. JCL to delete the required certificates is supplied in *artixhlq*.JCLLIB(DELCERT).

The GENCERT JCL creates the following data sets on your system:

- *artixhlq*.CERT.USERID.CA
- *artixhlq*.CERT.USERID.CLNT

## Step 2: Start the transformer service in SSL-enabled mode

To start the CICS transformer service in SSL-enabled mode:

1. First you must change the default configuration file so that it picks up the template for a secure configuration. By default, two configuration templates are provided in *artixhlq*.CONFIG:

| | |
|---|---|
| ARTIX | A default insecure (HTTP) configuration |
| AXSECURE | A default secure (HTTPS) configuration |

To switch to SSL-enabled mode, edit *artixhlq*.CONFIG(ARTIX) as follows to ensure that the include statement for AXSECURE at the end of the file is not preceded by a comment (#) character:

```
…
include "//artixhlq.CONFIG(AXSECURE)";
include "//artixhlq.CONFIG(AXINTRL)";
```

The AXSECURE file applies the required security settings by reopening the existing scopes in the ARTIX file.

2. Depending on the encryption strength on your machine, you might need to edit the configuration in *artixhlq*.CONFIG(AXSECURE). For example, if you have only export-strength encryption, edit the policies:mechanism_policy:ciphersuites variable so that it has the following value:

   ```
   "RSA_EXPORT_WITH_RC4_40_MD5"
   ```

3. Edit *artixhlq*.PROCLIB(ORXG), to add the System SSL library to your STEPLIB concatenation. The relevant line is shipped for your convenience as a comment, as follows:

   ```
   //* DD DISP=SHR,DSN=&SSLLOAD
   ```

4. Make sure the values for SSLHLQ and SSLLOAD are set in *artixhlq*.PROCLIB(ARTXVARS), as described in "Step 4: Set ITLOCALE (if necessary)" on page 37.
5. Submit one of *artixhlq*.JCLLIB(CICST), *artixhlq*.JCLLIB(CICSC), *artixhlq*.JCLLIB(IMST) or *artixhlq*.JCLLIB(IMSC) to start the transformer service you want to use.

### Step 3: Verify the transformer service is running

To verify that the transformer service is running:

1. Use FTP to copy the certificate in *artixhlq*.CERT.USERID.CLNT from z/OS to a Windows client machine. (Make sure you copy the file in binary format.)
2. Open Internet Explorer and select **Tools|Internet Options**.
3. Click the **Content** tab and then the **Certificates** button.
4. Click **Import** and then **Next**.

5. Click **Browse** to select the certificate file you copied from z/OS.
6. Import the selected file into your personal area. (The password to enter is `clientpass`.)
7. Check that your secured transformer service is running successfully.

   **Running in server mode:**
   If your secured transformer service is running in server mode, enter the following in a web browser from any client computer to confirm that the transformer service is ready to accept requests:

   ```
   https://remotehost:port/secureionasoap/
   ```

   In the preceding example, *remotehost* represents the z/OS TCP/IP hostname, and *port* represents the port on which the adapter is listening for client requests. (That is, port 12051 for the CICS transformer service, and 12050 for the IMS transformer service.) You should see a list of the services available to you in the transformer service. Select the service you require and you should see the generated WSDL for that service in your browser window.

   **Running in client mode:**
   If your secured transformer service is running in client mode, enter the following in a web browser from any client computer to confirm that the transformer service has started successfully and is accepting requests:

   ```
   https://remotehost:port/
   ```

   In the preceding example, *remotehost* represents the z/OS TCP/IP hostname, and `port` represents the port on which the transformer service is listening for deployments from Artix Designer.

### Step 4: Deploying to an SSL-enabled transformer service

When you want to deploy a project from Artix Designer to an SSL-enabled transformer service, you must select the **https** option on the **Deploy Project File on z/OS** window. For this to succeed, you must make sure that the CA certificate used by the mainframe-based transformer service has been made available to Artix Designer. For the details on accomplishing this, see the Artix Mainframe *Common User Tasks* guide.

## WebSphere MQ Customization Tasks

**Overview**

This subsection is only relevant if you want to enable the transformer service to use SOAP over WebSphere MQ. It describes the customization tasks to be performed before you can use the transformer service with WebSphere MQ.

**Step 1: Add the WebSphere MQ load libraries**

Edit *artixhlq*.PROCLIB(ORXG) to add the WebSphere MQ load libraries to your STEPLIB concatenation. The relevant lines are shipped for your convenience as comments, as follows:

```
//* DD DISP=SHR,DSN=&MQLOAD
//* DD DISP=SHR,DSN=&MQLANG
//* DD DISP=SHR,DSN=&MQAUTH
```

**Note:** If WebSphere MQ is not installed in the default z/OS location, edit the lines in *artixhlq*.PROCLIB(ARTXVARS) that define these variables, as appropriate for your site.

**Step 2: Start a WebSphere MQ-enabled transformer service**

To start a WebSphere MQ-enabled IMS or CICS transformer service in server mode, you must update the configuration file to indicate which Queue Manager and Queue Name the transformer service should listen to for requests.

To start a WebSphere MQ-enabled batch, CICS or IMS transformer service in client mode, there are no required changes to the supplied sample configuration.

For details on how to start the transformer service, see the user guide relevant to your chosen integration solution in the Artix Mainframe documentation library.

**Step 3: Grant any required security access to the transformer service**

WebSphere MQ allows many granularities of security settings. You can set access control on:

- Connections to WebSphere MQ
- WebSphere MQ objects, such as queues, processes, and namelists
- WebSphere MQ transmission links
- WebSphere MQ system control commands

- WebSphere MQ messages
- Context information associated with messages

To provide the necessary security, WebSphere MQ uses the z/OS system authorization facility (SAF) to route authorization requests to an External Security Manager (ESM) such as Resource Access Control Facility (RACF).

Depending on your site's requirements, you might need to grant the user ID running the transformer service access to various WebSphere MQ resources.

If you have command-specific security configured, be aware that the transformer service issues the following WebSphere MQ-specific calls: `MQCONN`, `MQOPEN`, `MQGET`, `MQPUT`, `MQPUT1`, `MQCLOSE`, and `MQDIS`. Specific queues you will be accessing may also be restricted.

Lastly, from a context perspective, the transformer service will update the `UserIdentfier`, `PutApplType`, `PutAppleName`, `PutDate`, `PutTime`, and the `ApplOriginData`. If you have context security enabled, you must grant the transformer service the necessary access to these contexts.

For more specific information on configuring security access, see Part 5, "Setting up Security" of the IBM publication *WebSphere MQ for z/OS System Setup Guide* (SC34-6052).

# Uninstalling Artix Mainframe z/OS Components

**Uninstalling the z/OS components**

To uninstall the z/OS-based components of Artix:

1. Stop all Artix services.
2. Delete all data sets under the high-level qualifier that you used for this installation.
3. Remove all system definitions that are no longer required. For example, remove ARTIX APF definitions from PARMLIB and from systems where these are currently active.
4. If you have been using the CICS or IMS transformer service in client mode, remove the reference to `artixhlq.MFA.LOADLIB` from your CICS or IMS regions accordingly.

# For More Information

**Release Notes**

See the Artix Mainframe *Release Notes* at:

```
http://www.iona.com/support/docs/artix/mainframe/5.1/release_notes/release_notes.pdf
```

**Knowledge base**

Review IONA Knowledge Base articles for Artix at:

```
http://www.iona.com/support/kb/index.jspa
```

**Technical support**

E-mail technical support with questions and suggestions at:

```
support@iona.com
```

# Index