Release V2R7.1
October 14, 2009

Cloud Compiler
Read This First

Program Product

Cloud Compiling

PROPRIETARY INFORMATION OF CLOUD COMPILING LLC

Table of Contents

Introduction

The Cloud Compiling™ Cloud Compiler™ (CC) is a family of cloud compiler licensed programs for the IBM OS/390 and z/OS operating environments. A cloud compiler is a program that functions equivalently to an actual compiler but does not require that the actual compiler be installed or licensed on the machine on which it runs. The CC utilizes FTP (the File Transfer Protocol client of the IBM OS/390 or z/OS SecureWay Communications Server) to transmit the user's source code to another mainframe (on which the actual compiler is installed), compile it there, and return the output of the actual compiler (system messages, listing, object code, etc.) to the user's specified target datasets. Most of the options and features of an actual compiler are supported.

This manual is a somewhat informal overview of the installation of one or more Cloud Compilers. Cloud Compiling publishes multiple cloud compiler program products and this Installation Guide applies equally to all of them. For more detailed installation information specific to each cloud compiler, and more formal feature by feature reference information, refer to the Cloud Compiling *Administration & Reference* manual for the specific cloud compiler.

The Cloud Compilers are not difficult to install. The primary tasks are the upload and TSO RECEIVE of a single load library on the machine on which compiles will be submitted, and the creation of a userid on the machine on which the IBM compilers will actually run. The SVCs are parametized with a small configuration file that may be edited under TSO. No IPLs, supervisor calls, authorized libraries, assemblies, or z/OS configuration changes are required – although certain of these steps (but never an IPL or the installation of a supervisor call!) may be indicated in some circumstances. Once basic management decisions have been completed and the userid has been defined, it should be possible to have the SVCs up and running in less than half a day. The installation process is well-documented, and unlimited telephone support is readily available.

Notices

See **Appendix I, Notices**.

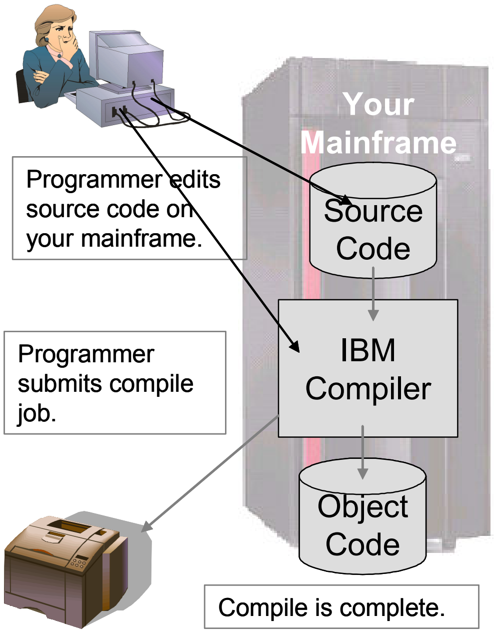
This manual was updated as of the date and software version number on the front cover but also applies to subsequent releases of the Cloud Compilers.

Technical Support – Contacting Cloud Compiling

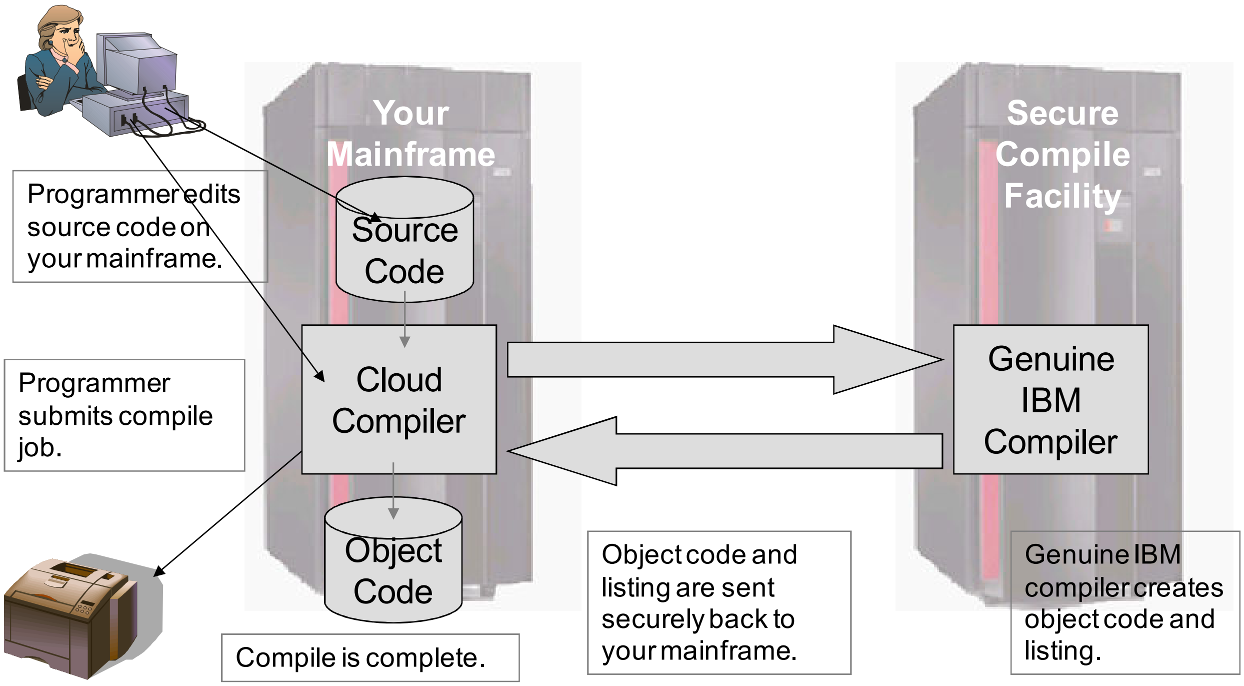
To contact Cloud Compiling LLC for technical support or for any other reason, please call 1-877-245-4322 and press 1, or send an e-mail to support@CloudCompiling.com.

Overview

How does a cloud compiler operate? Consider the following picture that shows the data flow in a typical compile.


Your Mainframe
Programmer edits source code on your mainframe.
Source Code
Programmer submits compile job.
IBM Compiler
Object Code
Compile is complete.

Now consider the following diagram, which shows how a cloud compile operates. Notice that from the application programmer's point of view nothing changes.


Your Mainframe
Secure Compile Facility
Programmer edits source code on your mainframe.
Source Code
Programmer submits compile job.
Cloud Compiler
Genuine IBM Compiler
Object Code
Compile is complete.
Object code and listing are sent securely back to your mainframe.
Genuine IBM compiler creates object code and listing.

This and all Cloud Compiling manuals use the term “server” to refer to the service mainframe on which the IBM compiler(s) is actually installed (the computer on the right in the picture above), and “client” to refer to the “user” mainframe on which a Cloud Compiler is used in place of one or more IBM compilers, and on which programmers submit compile jobs (the computer on the left). Do not draw any Windows or UNIX inference from the use of the term “server”; The IBM Terminology Web site (“Dictionary of Computing”) defines “server” as “(1) A software program or a computer that provides services to other software programs or other computers.”[1]

Management Decisions

There are three decisions that your local management must make – with your assistance and with assistance from Cloud Compiling’s technical support if they choose – that are prerequisites to the actual installation steps.

Choice of Machines

Your management must decide on which mainframe(s) to install the Cloud Compiler(s), and on which mainframe(s) to run the licensed IBM compilers. This is typically a financially-driven decision.

A corollary decision is whether to designate only a single mainframe as a compile server or to designate multiple machines so as to provide a backup capability. One fallback possibility is to leave the IBM compilers physically installed on your client machines, but to replace their usage with the cloud compilers and terminate your IBM compiler licenses. In the event of any problem with the compile server mainframe or dissatisfaction with the cloud compilers you may then fall back to the use of the IBM compilers with minimal disruption. You can even make that fallback automated and seamless through the use of the Sys-Safe feature, discussed in the installation manuals.

It is also possible to use the cloud compilers on all of your mainframes and use Cloud Compiling’s secure datacenter as a server for the licensed compilers, or as a backup server. Please contact your Cloud Compiling sales representative if you are interested in this option.

Authentication Method

For each cloud compile the client mainframe must authenticate itself (“sign on”) to the server mainframe. You have a choice of any (or a combination) of three authentication technologies: Passwords (referred to as KEYs in the *Administration & Reference* manuals), PassTickets, and NETRC files. Passwords is the easiest to implement; PassTickets is the most secure. (The use of NETRC is supported but not recommended.) You can also use one or a very small number of server machine userids for all compiles (recommended), or use each individual programmer’s userid on the server machine.

For a full discussion of these issues, see the section **Planning for FTP Server Authentication** in any of the Cloud Compiling *Administration & Reference* manuals.

[1] http://www-01.ibm.com/software/globalization/terminology/s.jsp#s18

Minimal JCL Changes – or None at All?

You need to decide which of two approaches to take to cloud compiling: do you want the convenience of absolutely no changes to compile JCL, or do you want to gain explicit control over cloud compiling by changing your compiler cataloged procedures (PROCs)?

As seen in the diagrams above, in any event the Cloud Compilers should be transparent to your application developers. The cloud compilers' inputs and outputs are nearly identical to the IBM compilers. However, there are several small differences. The cloud compiler reads one input file to determine parameters unique to cloud compiles such as the TCP/IP address of the compile server mainframe. The cloud compiler also creates three small listing files in addition to those created by the IBM compilers. Additionally, the cloud compiler *might* reside in a different load library than the IBM compiler, and *might* have a different entry point (PGM=) name.

Why change your compile JCL? If you decide to change your compile JCL you can specify the destinations for the three additional output files explicitly. You can specify the name of the parameter file in a DD statement, facilitating the use of several different parameter files. You can control explicitly whether a given job invokes the cloud or the IBM compiler.

Why choose "no JCL changes"? Perhaps there are too many cataloged procedures to change. Perhaps you do not have control over the programmers' JCL. Perhaps you wish to make the change to cloud compiling as transparent as possible for your programmers. *No matter what your ultimate decision, Cloud Compiling recommends that you take a "changed JCL" approach during your initial testing of the cloud compiler.*

You may also choose a combination of the two approaches. For example, you may decide to let the cloud compiler allocate its additional datasets without the use of DD statements, but you may decide not to change the entry point name to correspond to the actual compiler. For a full discussion of these issues and tradeoffs see **Cloud Compiling with No JCL Changes** and **JCL Changes for Maximum Control over Cloud Compiles** in any of the Cloud Compiling *Administration & Reference* manuals.

Setting Up the Server Mainframe(s)

Even if you will be installing multiple Cloud Compilers you will only need to do the steps in this section once.

There is no Cloud Compiling software *per se* that is installed on the server machine. The server mainframe must have the FTP “daemon” started task (usually named FTPD) installed, configured, and running. Of course, the compile server also needs all of the relevant IBM compilers, installed in the usual manner.

You will need one or more RACF[2] user IDs defined on the server mainframe. In Cloud Compiling’s experience this step often requires the cooperation of the security group, and so can consume days or weeks of elapsed time, so you may wish to start this process as soon as possible. If you will be using the “single user ID” method then you will need either one RACF user ID for each distinct programming “group” (however you wish to define “group” – possibly you will have only one) that will be using the SVCs. If you will be using the submitting programmers’ user IDs then presumably they are already defined in a shared RACF database. The CC user IDs may be defined using any naming convention you choose that is acceptable to RACF. The names must be seven characters or less because of the way that the CC generates job names (the usual convention of user ID plus one character). Cloud Compiling recommends that the userid be configured with no TSO logon privileges (to maximize security). If you will be using passwords for server authentication, Cloud Compiling recommends that the userid(s) be defined with a non-expiring password (to minimize CC administration). Neither of these options is absolutely necessary

You will need to select a job class to use for compiles, based on the initiators available on the server mainframe. You will need to select a held SYSOUT class for compile system messages. For JES3 systems the spool files must go to a hold queue reserved for external writers (a class for which HOLD=EXTWTR is specified in the JES3 installation stream).

If you will be using PassTickets for authentication, then you must perform the steps in **Server System PassTicket Configuration Tasks** in your *Administration & Reference* manual.

You may need more than one set of compiler defaults for a given compiler. This situation could occur, for example, if you are replacing a given compiler on more than one mainframe, and the groups using those two mainframes are accustomed to different compiler defaults. If you have this situation, then refer to **Appendix VI, Configuring Multiple Compiler Defaults** in the specific *Administration & Reference* manual.

In addition, there may be specific server system requirements for certain compilers, such as Java and Unicode support for the Enterprise COBOL compiler. Refer to your specific *Administration & Reference* manual for details.

[2] You may use RACF or another SAF-compliant z/OS security subsystem such as CA-ACF2 or CA-Top Secret. For simplicity, this document will use the term RACF to mean “your security subsystem.”

Setting Up the Client Mainframe(s)

The principal client mainframe task is actually installing the cloud compiler software. This process is documented in the sections **Preparing for Installation** and **Installation Steps** in the *Administration & Reference* manual. If you are installing more than one cloud compiler you will have to repeat these tasks for each cloud compiler.

If you decided to use PassTickets for server authentication, then perform the steps in **Client System PassTicket Configuration Tasks** in the *Administration & Reference* manual. If you decided to use passwords (keys), then perform the steps in **Appendix V, Configuring Keys**. If you are installing more than one cloud compiler you will have to do these steps only once.

You will need to tell the cloud compiler the IP address of the compile server mainframe, the name of the load library where the IBM compiler(s) is installed, and similar parameters. You do this by editing a small configuration file. Refer to the section **System Parameter File** in the *Administration & Reference* manual. If you are installing more than one cloud compiler, they can share a single System Parameter File – each cloud compiler processes the common parameters and its own COMPILER section, ignoring other COMPILER sections. There is a sample System Parameter File in the folder Control in the CC installation files. You will need to edit the first section of the sample file to specify the address of your compile server mainframe (either a name like SYSB.OURCORP.COM or a dotted address like 192.168.22.7), the userid you defined on the server, and either the masked password (KEY, if you are using passwords) or the FTP application name (PASSTICKET, if you are using PassTickets). You may need to edit the STEPLIB statements in the various COMPILER sections (or delete them or comment them out if your IBM compilers are linklisted on the server mainframe). Don't forget that the STEPLIB statement in the System Parameter File refers to the location of the IBM compiler(s) on the server mainframe, not the location of the cloud compiler on the client.

At this point you should be able to test your cloud compiler(s). Refer to **Operation** in the *Administration & Reference* manual. There is a sample test job included with your software installation. Start with a small and simple source program and work up to more complex tests. If you encounter problems – and problems are to be expected in any software installation – then refer to **Messages**. If you encounter timeouts, refer to **Appendix IV, Timeouts**. Please do not hesitate to contact Cloud Compiling technical support – see the section **Technical Support – Contacting Cloud Compiling** above. It is often worthwhile to specify VERBOSE YES in the System Parameter File and re-run the failing test before calling Cloud Compiling Technical Support. If you are installing more than one cloud compiler then of course you should test all of them.

Progress to larger test compiles. You may find that you need to increase timeouts or the size of various compiler files. See **Appendix IV, Timeouts** in the *Administration & Reference* manual if you have a problem with timeouts on larger compiles. See the descriptions of the various "SPACE" parameters (ADATASPACE, COPYPDSSPACE, etc.) in the section **Appendix II, Cloud Compiler System Parameters** if you encounter out-of-space conditions or Sx37 abends.

If you are using a "no JCL changes" approach you will probably want to build a module that identifies the name of the default System Parameter File. See **Setting the Default System Parameter File Name** for instructions.

In some cases you may be able to improve the performance of the cloud compilers by APF-authorizing the load library in which they are installed (although APF-authorization is *never required*). See **Authorizing the CC Load Libraries**.

If you are interested in detailed z/OS accounting for cloud compiler-submitted jobs on your server mainframe refer to the section **Server-Side Job Accounting** in the *Administration & Reference* manual.

Putting Your Cloud Compiler(s) into Production

Cloud Compiling recommends that you begin with a single application developer. Create a custom cataloged procedure for him or her to work with for a day or two.

Then expand your testing to one or two development groups.

Finally, put the cloud compiler into production for you entire organization. Either modify the main compile cataloged procedures, or alias the cloud compiler so that all references to the IBM compiler invoke the cloud compiler. See **Assigning the IBM Compiler Name as an CC Alias** in the *Administration & Reference* manual.

If you are installing more than one cloud compiler you should repeat these steps for each cloud compiler, although it may be safe to accelerate the process for the second and subsequent compilers.

Your Cloud Compiler implementation is complete.

Installation Checklist

Bold face type indicates the name of a relevant section in your *Administration & Reference* manual.

Management Decisions

□ Decide on which mainframe(s) to install the Cloud Compiler(s) and on which mainframe(s) to run the licensed IBM compilers.

□ Decide whether to utilize the Sys-Safe feature.

□ Choose an authentication method, passwords (keys), PassTickets, or NETRC. **Planning for FTP Server Authentication**.

□ Minimal JCL Changes – or None at All? **Cloud Compiling with No JCL Changes** and **JCL Changes for Maximum Control over Cloud Compiles**.

Setting Up the Server Mainframe(s)

Even if you will be installing multiple Cloud Compilers you will only need to do the steps in this section once.

□ Define one or more RACF user IDs on the server mainframe.

□ Select a job class to use for compiles.

□ If you will be using PassTickets, perform the steps in **Server System PassTicket Configuration Tasks**.

□ If you need more than one set of compiler defaults for a given compiler then refer to **Appendix VI, Configuring Multiple Compiler Defaults**.

□ Any server system requirements for certain compilers, such as Java and Unicode support for the Enterprise COBOL compiler.

Setting Up the Client Mainframe(s)

Even if you will be installing multiple Cloud Compilers you will only need to do the steps in this section once *except as otherwise noted*.

□ Define a load library and upload the cloud compiler software. **Preparing for Installation** and **Installation**. Repeat for each cloud compiler you are installing.

□ If you will be using PassTickets for authentication then perform the steps in **Client System PassTicket Configuration Tasks**.

□ If using passwords (keys) for authentication then perform the steps in **Appendix V, Configuring Keys**.

□ Customize the System Parameter File. **System Parameter File**.

□ Initial testing of your cloud compiler. **Operation**. Repeat for each cloud compiler you are installing. Test the Sys-Safe feature, if utilized.

□ If you chose “no JCL changes” build a module that identifies the default System Parameter File. **Setting the Default System Parameter File Name**.

□ APF-authorize the CC load library(ies) if indicated. **Authorizing the CC Load Libraries**.

□ If you are interested in detailed accounting for cloud compiler-submitted jobs refer to **Server-Side Job Accounting**.

Putting Your Cloud Compiler(s) into Production

If you are installing more than one cloud compiler you should repeat these steps for each cloud compiler

□ Testing with a single application developer.

□ Expanded testing with one or two development groups.

□ Put the cloud compiler into production for you entire organization. Modify the main compile cataloged procedures or alias the cloud compiler to the name of the IBM compiler. **Assigning the IBM Compiler Name as an CC Alias**.

Your Cloud Compiler implementation is complete.

Appendix I, Notices

Cloud Compiling may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents.

CLOUD COMPILING PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Cloud Compiling may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Trademarks

Cloud Compiling, Sys-Safe and Cloud Compiler are trademarks of Cloud Compiling LLC.

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM®
COBOL/370
eServer
MVS
MVS/ESA
OS/390®
RACF
S/390®
z/OS®
zSeries®

CA-ACF2®, CA-Librarian®, CA-Optimizer®, and CA-Top Secret® are registered trademarks of CA, Inc.

Compuware, Compuware Shared Services and Xpediter are trademarks of Compuware Corporation.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc.

Macintosh® is a registered trademark of Apple Computer, Inc.

UNIX® is a registered trademark of The Open Group.

Windows® is a registered trademark of Microsoft Corporation.

Other company, product, or service names may be trademarks or service marks of others. No association with Cloud Compiling is implied.