

*Master building accurate z/OS applications in less time with less effort*

OpenMake Meister™ for z/OS is the only solution available that allows you to build and manage binaries, across the enterprise - from Windows/Unix to the Mainframe, providing a “one size fits all” solution.

Meister for z/OS provides a central, common method for managing your binaries, regardless of the language or Operating System. So go ahead, develop on the mainframe, we have you covered.

## Key Benefits

- Synchronizes build to release between z/OS and distributed systems.
- Supports z/OS build Best Practices and Project Management
- Exposes build forensics that link a production binary back to its development origins, with or without Source Code Management.
- Advanced support of mainframe development languages including CICS/DB2, Assembler and PL1
- On demand access to critical build-to-release information via the centralized, community developed knowledge base.
- Enables collaboration and cooperation between z/OS developers and distributed developers

```
-----Setup Build-----
Row 1 to 9 of 23
COMMAND ==>
Scroll ==> CSR
  Build Highlevel:  P390A.OPENMAKE.OOMSAMPL

  Project          ==>  MAINFRAME BUILD
DEMO              / to Choose
  Search Path ==>  RE-
LEASE2           / to Choose

  Bldmake Options / to Choose
  OM Options      / to Choose

  Actions: (A)dd, (C)opy, (D)elete, (E)dit, (B)uild and Build
(All)

  Target          OS      Build Type
load(asmhi)      zOS    Assembler Load Module
load(cblhi)      zOS    COBOL Load Module
load(cblhib2)    zOS    DB2 COBOL Load Module
load(cicimscb)   zOS    CICS IMS COBOL Load Module
load(cmdprs00)   zOS    Assembler Obj Load Module
load(db2cobci)   zOS    DB2 CICS COBOL Load Module
load(hic)        zOS    C Load Module
load(hicpp)      zOS    CPP Load Module
load(hicppdb2)   zOS    DB2 C Load Module
```

**z/OS Project with diverse Build Methods**

## Build Best Practices: Meister's Build Methods

Meister for z/OS Build Methods enables build best practices by defining standard procedures for calling z/OS compilers. Build Methods are based on REXX and can be developed using Meister for z/OS's rich library of reusable REXX modules that perform common tasks such as generating JCL. Build Methods enables you to generate the Compile and Links JCL, in a consistent and repeatable method, minimizing the redundancies normally found in ad hoc JCL scripting. Build methods enable build best practices and enforces standard procedures in every build you execute.

## Adaptable and Dynamic Builds: Meister Build Services

OpenMake Meister for z/OS is the perfect solution for developers writing applications that connect diverse platforms. When your application spans both the legacy mainframe environment as well as the distributed environment, you need a method for building the application as if it runs on a single platform. Meister's Build Services standardizes and centralizes the creation of binaries for both the mainframe and distributed platforms. Meister allows you to integrate your environments by providing a common process for building your application and managing your binaries.

## On-demand Build Knowledge: Meister's Community Developed Knowledge Base

Meister for z/OS integrates into the Meister community developed knowledge base enabling cross platform build coordination. The community developed knowledge base keeps all developers in sync providing a shared build object repository, common locations for reporting on parameter usage, and environment variable usage. Build Logs, reports and schedules are centralized providing the self-service and self-documented support needed in today's lean development environments.

## Build Forensics: Managing Source to Load Parity

Meister for z/OS provides the deep dependency discovery needed for exposing the artifacts used to create your load objects. Even when components are NOT stored in a version control tool, Meister for z/OS will provide the precise location of where all source, objects and libraries were found to create the binary. When an application failure occurs you need to trace the source of the offending event, similar to using fingerprints and DNA to track down the suspect of a crime. With Meister for z/OS advanced dependency discovery and reporting, every item used to create the binary is stored in an application "footprint" and reported in the applications "Build Audit Report" regardless of versioning tool usage.

**For sales information and demo request, call:**  
**Toll Free: 800.359.8049**  
**HQ: 312.440.9454**  
**North American Sales: 805.696.6866**

## Meister for Java Feature Summary

z/OS Load Module Management	✓
Source to Load Build Forensic with binary footprints	✓
JCL Consolidation through Build Services and Methods	✓
Build best practices and Project Management for z/OS	✓
Flexible Build Project Management supporting any code organization	✓
Synchronization between z/OS builds and distributed builds	✓
Incremental Builds drastically reducing build times	✓
Self-documented Builds	✓
Configuration Parameter Management (Debug Flags, Optimization Parameters, etc)	✓
Directory Management	✓
Variable Management	✓
Dependency Discovery	
File to File Relationship scanning	
Project to Project Relationship scanning	
Project to 3 <sup>rd</sup> Party Object Relationship scanning	✓
Build Dependency Audit Report without reliance on SCM repository	✓
Build Audit Logs exposing Environmental Configurations for Build Repeatability	✓
Configuration Parameter Usage Reports	✓
Dependency Usage Reports	✓
Impact Analysis Implosion and Explosion Reports	✓
Mojo Process Automation and Workflow Management	✓
Centralized "Smart" Logging with easy to read result summaries and error warnings	✓
User and Group Privileges with LDAP support	✓
Accelerated Cross Platform and Distributed Builds	✓
Application Programming Interface (API)	✓
Remote site support with Remote Build Servers	✓
Remote Build Server catalog	✓
Market Acceptance with 13 years of proven field experience supporting over 200 development environments	✓
Established ALM partner relationships with IBM, CA, MKS, Serena, Perforce, and AccuRev	✓



213 W. Institute Place, #404  
 Chicago, IL 60610  
 Tel: 312.440.9545  
 Toll Free 800.359.8049  
 Fax: 312-440-9543