

Mundi 엔진 심볼 서버 (Symbol Server)

Mundi 엔진의 PDB 심볼 파일을 중앙 집중식으로 관리하고, Git 소스 인덱싱을 통해 디버깅 효율성을 향상시키는 시스템입니다.

📄 목차

- [개요](#)
- [시스템 요구사항](#)
- [초기 설정](#)
- [사용 방법](#)
- [Visual Studio 디버거 설정](#)
- [문제 해결](#)
- [파일 구조](#)

개요

심볼 서버란?

심볼 서버는 소프트웨어 디버깅에 필요한 **PDB(Program Database) 파일**을 중앙에서 관리하는 시스템입니다.

주요 기능:

- 🕒 **자동 심볼 로드**: 디버거가 크래시 덤프 분석 시 자동으로 PDB 다운로드
- 📁 **버전 관리**: 모든 빌드 버전의 심볼 파일 보관
- 🌐 **소스 인덱싱**: Git 커밋 정보 기반 소스 코드 자동 다운로드
- 👥 **팀 협업**: 팀 전체가 동일한 심볼 서버 공유

작동 원리

빌드 시:

Mundi.exe + Mundi.pdb 생성
↓
Git 정보 삽입 (커밋, 브랜치, remote URL)
↓
심볼 서버에 업로드 (symstore.exe)

디버깅 시:

크래시 덤프 파일 오픈
↓
디버거가 심볼 서버에서 PDB 자동 다운로드
↓
PDB 내 Git 정보로 소스 코드 자동 다운로드
↓
정확한 함수명 + 소스 코드 줄 번호 표시

시스템 요구사항

필수 소프트웨어

도구	용도	다운로드
Windows Debugging Tools	srctool.exe, pdbstr.exe, symstore.exe	Windows SDK
Python 3.x	소스 인덱싱 스크립트 실행	Python.org
Git	소스 버전 관리	Git-SCM
PowerShell 5.1+	배포 스크립트 실행	Windows 기본 포함

Windows Debugging Tools 설치

1. **Windows SDK 설치 관리자** 다운로드
2. 설치 옵션에서 **"Debugging Tools for Windows"** 선택
3. 기본 설치 경로: `C:\Program Files (x86)\Windows Kits\10\Debuggers\x64`

설치 확인:

```
Test-Path "C:\Program Files (x86)\Windows Kits\10\Debuggers\x64\symstore.exe"
```

초기 설정

1단계: 네트워크 공유 폴더 설정

심볼 서버로 사용할 네트워크 공유 폴더를 생성합니다.

서버 PC에서 실행 (관리자 권한):

```
# 심볼 저장소 디렉토리 생성
New-Item -ItemType Directory -Force -Path "C:\SymbolServer\Symbols"

# SMB 공유 생성
New-SmbShare -Name "MundiSymbols" `
  -Path "C:\SymbolServer" `
  -ReadAccess "Everyone" `
  -FullAccess "Administrators"

# 공유 확인
Get-SmbShare -Name "MundiSymbols"
```

접근 경로:

- UNC 경로: `\\[서버이름 또는 IP]\MundiSymbols\Symbols`
- 예시: `\\192.168.1.100\MundiSymbols\Symbols`

2단계: 설정 파일 수정

`SymbolServerConfig.json` 파일을 편집합니다:

```
{
  "symbolServer": {
    "path": "\\YOUR_SERVER_IP\MundiSymbols\Symbols", // ← 여기를 실제 경로로
    변경
    ...
  }
}
```

변경 예시:

```
{
  "symbolServer": {
    "path": "\\192.168.1.100\MundiSymbols\Symbols",
    ...
  }
}
```

3단계: Windows Debugging Tools 경로 확인

기본 경로가 아닌 경우 `SymbolServerConfig.json`에서 수정:

```
{
  "winDbgKit": {
    "path": "C:\\Program Files (x86)\\Windows Kits\\10\\Debuggers\\x64"
  }
}
```

사용 방법

자동 배포 (빌드 후 자동 실행)

`Mundi.vcxproj`의 `PostBuildEvent`에 포함되어 있어 **빌드 완료 시 자동으로 심볼이 배포됩니다.**

적용된 구성:

- Debug
- Release
- Debug_StandAlone
- Release_StandAlone

빌드만 하면 자동으로 심볼 서버에 업로드됩니다!

수동 배포

빌드 없이 기존 PDB를 수동으로 배포하려면:

방법 1: 배치 파일 실행 (간편)

```
# Release 구성 배포
PublishSymbols.bat Release

# Debug 구성 배포
PublishSymbols.bat Debug
```

또는 파일 탐색기에서 `PublishSymbols.bat`를 더블 클릭 (기본값: Release)

방법 2: PowerShell 직접 실행

```
# Release 구성
.\IndexAndPublish.ps1 -Configuration Release

# Debug 구성 (소스 인덱싱 제외)
.\IndexAndPublish.ps1 -Configuration Debug -SkipSourceIndex
```

배포 결과 확인

성공 시 다음과 같은 메시지가 출력됩니다:

```
=====
Symbol Server Publish Complete
=====
Configuration:  Release
PDB File:       C:\...\Binaries\Release\Mundi.pdb
Symbol Server:  \\192.168.1.100\MundiSymbols\Symbols
Git Commit:     abc123def456...
Git Branch:     feature/Symbol-Server
Version:        Release-feature/Symbol-Server-20251122-143025
=====
```

Visual Studio 디버거 설정

Mundi 엔진은 **자동으로 심볼 서버 경로를 설정**하므로, 최소한의 수동 설정만 필요합니다.

자동 설정 항목 (Mundi 엔진이 자동 처리)

☒ 심볼 서버 경로 (`_NT_SYMBOL_PATH` 환경 변수) ☒ 소스 서버 지원 (`_NT_SOURCE_PATH` 환경 변수)

작동 방식:

- Mundi.exe를 **한 번만 실행**하면 Windows 레지스트리에 환경 변수가 자동 설정됩니다
- 모든 디버거(Visual Studio, WinDbg, Rider 등)가 자동으로 이 설정을 인식합니다
- Visual Studio를 **재시작**하면 설정이 적용됩니다

수동 설정 항목 (사용자가 1회 설정 필요)

모든 모듈 기호 검색 활성화

1. Visual Studio 메뉴: **도구** → **옵션**
2. **디버깅** → **기호** 선택
3. **기호 검색 기본 설정** 섹션에서:
 - ☒ "제외되지 않은 모든 모듈 기호 검색(A)" 선택
 - ☒ "검색할 모듈 기호를 자동으로 선택(C)" 선택 해제

이 설정이 필요한 이유:

- 기본 설정("자동으로 선택")은 로드된 모듈만 심볼을 검색합니다
- 덤프 파일 디버깅 시에는 모든 모듈의 심볼이 필요하므로 "모든 모듈 검색"으로 변경해야 합니다

완료!

이제 크래시 덤프 파일을 열면:

- ☒ PDB가 자동으로 다운로드 (심볼 서버에서)
- ☒ 소스 코드가 자동으로 다운로드 (Git에서)
- ☒ 함수명과 줄 번호가 정확히 표시

(선택사항) 추가 설정

Microsoft 심볼 서버 추가

Windows DLL 디버깅을 위해 Microsoft 심볼 서버도 추가 가능:

1. **도구** → **옵션** → **디버깅** → **기호**
2. **기호 파일(.pdb) 위치** 섹션에서 **새 위치 추가**
3. 경로 입력:

```
https://msdl.microsoft.com/download/symbols
```

참고: Mundi 엔진은 이미 Microsoft 심볼 서버 경로를 자동 설정하므로, 이 단계는 선택사항입니다

문제 해결

문제: "PDB file not found" 오류

원인: 빌드 후 PDB 파일이 생성되지 않음

해결:

1. Visual Studio 프로젝트 속성 확인
2. **C/C++** → **일반** → **디버그 정보 형식**: **/ZI** 또는 **/ZI** 설정
3. **링커** → **디버깅** → **디버그 정보 생성**: **예** 설정

문제: "src tool.exe not found" 오류

원인: Windows Debugging Tools 미설치 또는 경로 오류

해결:

1. Windows SDK 설치 확인
2. **SymbolServerConfig.json**에서 경로 확인:

```
"winDbgKit": {  
  "path": "C:\\Program Files (x86)\\Windows Kits\\10\\Debuggers\\x64"  
}
```

문제: "Symbol server path not configured" 오류

원인: 설정 파일이 기본값 상태

해결: **SymbolServerConfig.json** 편집:

```
{  
  "symbolServer": {  
    "path": "\\YOUR_SERVER_IP\\MundiSymbols\\Symbols" // ← 실제 경로로 변경  
  }  
}
```

문제: 네트워크 공유 접근 거부

원인: SMB 공유 권한 문제

해결:

```
# 공유 권한 확인  
Get-SmbShare -Name "MundiSymbols" | Get-SmbShareAccess  
  
# 권한 추가 (관리자 권한)  
Grant-SmbShareAccess -Name "MundiSymbols" -AccountName "Everyone" -AccessRight  
Read -Force
```

문제: Git 정보를 가져올 수 없음

원인: Git 저장소가 아니거나 Git 미설치

해결:

- Git 설치 확인: `git --version`
- Git 저장소 확인: 프로젝트 루트에 `.git` 폴더 존재 확인

문제: 소스 코드 자동 다운로드 실패

원인: Private repository 접근 권한 없음

해결:

1. Git credential manager 설정
2. SSH 키 또는 Personal Access Token 등록
3. 또는 소스 인덱싱 건너뛰기:

```
.\IndexAndPublish.ps1 -Configuration Release -SkipSourceIndex
```

파일 구조

Mundi\BuildTools\SymbolServer\

|

├─ SymbolServerConfig.json

├─ IndexAndPublish.ps1

├─ SourceIndex.py

├─ PublishSymbols.bat

└─ README.md

설정 파일 (심볼 서버 경로 등)

메인 배포 스크립트 (PowerShell)

Git 소스 인덱싱 스크립트 (Python)

수동 실행 래퍼 (Batch)

사용 가이드 (이 문서)

각 파일의 역할

파일	역할	언제 사용?
SymbolServerConfig.json	심볼 서버 경로, Debugging Tools 경로 설정	초기 설정 시 1회 편집
IndexAndPublish.ps1	PDB 소스 인덱싱 및 심볼 서버 배포 자동화	빌드 후 자동 실행 또는 수동 실행
SourceIndex.py	Git 정보로 Source Server 스트림 생성	IndexAndPublish.ps1이 자동 호출
PublishSymbols.bat	간편한 수동 배포 인터페이스	수동 배포 시 더블 클릭
README.md	사용 설명서	도움이 필요할 때 참고

고급 사용법

특정 파일만 배포

```
# symstore.exe 직접 사용
& "C:\Program Files (x86)\Windows Kits\10\Debuggers\x64\symstore.exe" `
    add /f "C:\...\Binaries\Release\Mundi.pdb" `
    /s "\\192.168.1.100\MundiSymbols\Symbols" `
    /t "Mundi" `
    /v "Manual-Upload" `
    /c "Manual upload by user"
```

오래된 심볼 삭제

심볼 서버 용량 관리를 위해 오래된 심볼을 삭제:

```
# 90일 이상 된 심볼 삭제
$SymbolPath = "\\192.168.1.100\MundiSymbols\Symbols"
$DaysToKeep = 90

Get-ChildItem $SymbolPath -Recurse |
    Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-$DaysToKeep) } |
    Remove-Item -Force -Recurse
```

CI/CD 통합 (GitHub Actions 예시)

```
- name: Publish Symbols
  run: |
    cd Mundi\BuildTools\SymbolServer
    .\IndexAndPublish.ps1 -Configuration Release
  if: github.ref == 'refs/heads/main'
```

참고 자료

- [Microsoft Docs: Symbol Server](#)
- [Debugging Tools for Windows](#)
- [Source Server 개요](#)

라이선스 및 기여

이 심볼 서버 시스템은 Mundi 엔진 프로젝트의 일부입니다.

문의 및 버그 리포트:

- 프로젝트 이슈 트래커에 등록해주세요

Happy Debugging! 🐛🔍