MOZART Management Console

MMC 사용자 가이드 목차

Introduction /mmc-introduction
 Server & Client Settings /server-and-client-settings
 MMC User Guide /mmc-user-guide

Introduction

MMC 개요

MOZART Management Console (이하 MMC) 는 MOZART IDE 를 통해 개발된 작업이나 이메 일 발송, 프로그램 실행과 같은 사전 정의된 작업을 특정 시간, 특정 이벤트의 발생, 종료 조건 에 따라 수행할 수 있도록 작업 실행 예약처리를 할 수 있는 Job Scheduling 기능과 서버에서 실행할 수 있는 작업을 등록하기 위한 작업의 등록 및 실행파일 업로드, 이력관리 등의 배포 관리 기능으로 구성된 MOZART Server 관리 도구입니다.

Mozart Job Scheduling 주요 개념

Job Scheduling 은 Job Type(작업 유형), Job(작업), Trigger(트리거) 의 세 가지 관리 항목으 로 구성됩니다.

Job Type

Job Type 은 Job Scheduler 에 의해 실행될 수 있는 작업 유형입니다. 작업 유형에는 일반적 인 전자 메일 메시지 보내기 및 프로그램 실행 유형과 MOZART 에 의해 개발된 Simple Task, Model Task, Collaboration Task의 세 가지 유형이 있습니다. 이 중 Model Task 나 Collaboration Task 의 경우에는 대상 Model (Model의 개념 설명 참조)과 인수의 설정에 따라 서만 실행 내용이 변경되므로, Job Type 이 사전에 정의되어 있습니다. 신규 정의가 필요한 Job Type 은 Model 에 기반을 두지 않는 Simple Task 에 국한됩니다. (Job Type 관리방법)

Job

Job 은 Job Type(작업유형)의 Parameters 가 설정된 실행 대상 작업으로 Trigger 에 의해 실 행될 수 있습니다. JobType 은 Arguments 설정에 따라 다르게 동작하는 여러 개의 Jobs 으 로 설정이 가능합니다. 즉, Job Type 과 Job 은 1 : N 의 관계를 가집니다. 예를 들어 전자 메일 메시지 보내기 작업유형으로 보내는 이, 받는 이, 내용 설정 등의 Arguments 를 달리하여 복 수 개의 전자 메일 발송 작업을 생성할 수 있습니다. (Job 관리 방법)

Trigger

Trigger 는 대상이 되는 작업을 실행하기 위한 조건과 실행 방법을 정의한 정보의 집합입니다. Trigger 는 실행할 대상 작업과 작업을 실행하기 위한 조건을 정의하여 생성할 수 있습니다. 작업은 MMC에 의해 등록된 Job 이어야 하며 작업조건은 시간 기반 방법(Time Scheduler)과 이벤트 기반(Condition)의 두 가지 방식으로 정의할 수 있습니다. (Trigger 관리 방법)

- 시간 기반 조건(Time Scheduler) : 작업을 실행하기 위해 특정 시점, 수행 주기 등을 정의 하는 방식
- 이벤트 기반 조건(Condition) : 다른 Trigger 의 시작, 종료 이벤트에 의해 작업을 실행하 도록 정의하는 방식

아래 그림은 각 주요 개념의 관계를 도식화한 것입니다.



MOZART Model 개요

MOZART Model은 MOZART 로 개발된 Logic 에서 사용되는 Input/Output Data Schema, Query, Data 연결정보 및 Logic 의 처리 시에 활용되는 Arguments 에 대한 정의를 포함하는 Data 집합입니다. Logic 의 개발 시점에는 Schema 와 Arguments 가 사용되며 실행시점에 Data 연결정보와 Query 를 사용해 Data 를 받아오거나 저장하는 작업을 수행할 수 있습니다. 특히 실행을 위해서는 해당 Model 을 실행할 Logic 이 포함된 Assembly 정보와의 연결 정보 가 필요하여 이 정보 또한 Model 에 포함되어 있습니다. 즉 MOZART Model 을 실행하기 위해 서는 Model 파일과 Model 을 실행하기 위한 실행파일이 필요합니다.

MOZART Framework 상에서 이러한 Model 을 실행하는 방법은 크게 두 가지 입니다. 첫 번째 는 MOZART Studio 를 통해 실행하는 방법입니다. 보통 개발자가 개발 중 테스트를 위해 Model 을 실행하거나, 동일한 Model 에 대해 입력정보를 변경하여 다양한 실험을 수행해보 는 경우에 주로 사용하는 방법입니다. 두 번째는 MOZART Server에 의해 실행하는 방법입니 다. Model 실행 결과를 사용 시나리오에 따라 운영에 적용하기 위해 사용하는 방법입니다. 아 래 그림은 Model 의 구성과 실행구조를 설명합니다.



Model Task 는 MOZART 에서 기본으로 제공하는 Job Type(작업유형) 으로 실행 대상 Model 과 Model Task 에 의해 수행되는 (0)~(3)까지의 작업 수행방식을 설정하기 위한 설정값 (Extended Arguments) 을 입력하여 대상 Model 을 실행시켜주는 역할을 합니다. MOZART Studio 는 Model Task 와 유사하게 Model 을 실행시킬 수 있으나, 기본적으로 Inputs 이 생성 된 상태의 Model 을 실행시키는 경우가 많으므로 Logic 실행 Input Data 를 Download 받는 작업은 필요 시 별도의 메뉴로 실행하며, Model 실행은 (1)~(3)까지 작업을 묶어 실행하게 됩 니다.

작업 유형 (Job Type)

Job Type 은 Job Scheduler 에 의해 실행될 수 있는 작업유형입니다. Job Type 은 실행을 위 해 실행 방법을 결정하는 Arguments 를 가집니다. Job Type 은 Job 관리 화면의 **Edit Job Type** 메뉴에 의해 관리가 가능합니다. 또한 Job 을 정의할 때 선택하도록 표시됩니다.

다음은 MOZART Job Scheduler가 제공하는 기본 작업유형에 대한 설명입니다.

전자 메일 보내기(\$sendmail)

해당 Job Type 으로 생성된 작업(Job) 이 실행되면 전자 메일을 보냅니다. Job Type 의 Arguments 로 보낸 사람, 받는 사람, 제목, 본문 및 첨부파일을 지정해야 하며, 보내는 메일 Server (SMTP) 설정을 해주어야 합니다. 관리목적으로 MMC 에 의해 실행되는 특정 Job 수 행이 실패하는 등의 경우에 메일을 발송하도록 할 수 있습니다.

프로그램 실행(\$exec)

프로그램이나 스크립트를 시작합니다. Program/script 에서 지정된 프로그램을 실행하며 지 정된 프로그램/스크립트에 명령줄 인수가 사용되는 경우 Add arguments(optional) 텍스트 상자에 이러한 인수를 제공할 수 있습니다. Start in (optional) 텍스트 상자에서 프로그램이 나 스크립트를 실행하는 명령줄에 작업 디렉터리 (Working Directory) 를 지정할 수 있습니다. 이 디렉터리는 프로그램이나 스크립트 파일의 경로이거나 실행 파일에 사용되는 파일 경로 여야 합니다. Windows 에서 기본적으로 제공하는 프로그램 및 사용자가 작성한 일반 실행파 일은 권한이 있는 경우 모두 실행 가능합니다.

Model Task (\$model)

Model Task 는 MOZART IDE 를 통해 개발된 Model 기반 작업을 실행하기 위한 작업 유형입 니다. Model Task 작업유형은 Model 의 구동 방식(Mozart Model 개요 참조)을 설정하기 위 한 기본 Arguments (Extended Argument 참조) 를 가지고 있습니다. 가장 기본이 되는 Arguments 는 Model Task 가 실행할 Model 정보로 Server 의 Working Folder 에 저장된 Model 을 지정합니다. 실행 대상이 되는 Model 은 수행을 위해 설정되어야 할 내용을 내부 Arguments 로 정의하고 있어 작업을 정의할 때 이 Arguments 들도 설정이 되어야 합니다.

Collaboration Task (\$cola)

Collaboration Task 의 경우는 복수 개의 작업을 실행 시점에 작업 간의 통신을 통해 협응하 여 실행할 수 있도록 만든 작업입니다. 기본이 되는 Model 을 선택하고 해당 Model 과 Collaboration 대상이 되는 Model 을 설정해 주어야 합니다. (Extended Argument 참조)

Job

Job 은 Job Type(작업 유형) 의 Parameters 가 설정된 실행 대상 작업으로 Trigger 에 의해 실 행됩니다. Job Type 은 Arguments 설정에 따라 다르게 동작하는 여러 개의 Job 으로 설정이 가능합니다. 즉, Job Type 과 Job 은 1 : N 의 관계입니다.

Job 을 정의하는 정보는 아래와 같습니다.

Basic Information

- Job Name : Job 명칭입니다. Job 명은 Job 을 유일하게 구분하는 구분자로 사용됩니다.
- **Description** : Job 에 대한 설명입니다.
- Job Type : Job 이 실행할 Job Type 입니다. 사전 정의된 Job Type(Job Type 참조) 과 사 용자가 등록한 Job Type 중에서 선택하여 지정할 수 있으며 선택된 Job Type 에 따라 설 정할 수 있는 Arguments 가 변경됩니다. Job Type 중 Model(Model 개념 참조)을 실행하 도록 하는 Model Task, Collaboration Task 의 경우 추가적인 속성을 선택할 수 있습니다. Model 기반의 작업을 실행하기 위해서는 기본적으로 Model 이 지정되어야 하며, Model 의 실행 방식과 협업 방식에 따라 아래 추가적인 속성의 설정이 필요합니다.
 - Model file : Model Task 가 실행할 Model File 의 경로입니다. 서버의 Working Folder 를 Base Folder 로 하여 설정합니다.
 - Model dll file : Job 수행을 위해 개발된 DII 파일을 지정합니다.
 - Log dir : 해당 Job 의 수행 Log 를 남길 Folder 를 지정합니다. Working Folder 의 Logs Folder 를 Base Folder 로 하여 지정합니다.
 - Additional run count (Optional) : Job 의 추가 실행 횟수를 지정합니다. '0' 인 경우 추가 실행을 하지 않고 1번만 실행하며, '1' 인 경우 총 2회를 실행하게 됩니다. (More Run 설정 방법 참조)
 - Collaboration Count (Optional): Job 실행 시 다른 Job 과 협업이 필요한 경우 대상 Job 의 수를 의미합니다. Collaboration Task 의 경우 여러 개의 Job 이 수행 중에 통 신하며 협업할 수 있도록 지원하며 본 속성은 협업 대상 Job 의 수를 의미합니다. 설 정 수만큼 협업 대상 Job 을 지정할 수 있습니다. (Collaboration 설정 방법)
- Disallow Concurrent Execution : Job 동시 실행 가능 여부에 대한 옵션입니다. 해당 옵션 을 사용(체크)하는 경우에는 Job 이 여러 Trigger 에 동시에 계획되어도 동시에 실행이 불가합니다.

Arguments

- Job Type 별 Arguments
- 설정된 Job Type 의 Arguments 입니다. 전자 메일 보내기 작업 유형의 경우 보내는 사람, 받는 사람, 제목, 내용 등이 Arguments 가 되며, 사용자가 정의한 Job Type 의 경우 Task 개발 시 설정을 위해 만든 Arguments 가 이에 해당합니다. Model Task 와 Collaboration Task 에서 사용하는 사전 정의된 Arguments(Extended Arguments 참조) 도 Job Type 별 Arguments 에 해당합니다.
- Model Arguments
- Model Task, Collaboration Task에서 실행 대상으로 설정된 Model 에 정의된 Arguments 입니다. Job 에 설정된 Arguments 별 값들은 Trigger 를 생성하는 시점에 Parameters 기 본값으로 사용됩니다. 다만, Trigger 에서 Parameters 의 값을 재정의하면 Trigger 에서 설 정한 값이 실행 시 반영됩니다.

Job 은 Job Type(작업 유형) 의 Parameters 가 설정된 실행 대상 작업으로 Trigger 에 의 해 실행됩니다. Job Type 은 Arguments 설정에 따라 다르게 동작하는 여러 개의 Job 으 로 설정이 가능합니다. 즉, Job Type 과 Job 은 1 : N 의 관계입니다. Job 을 정의하는 정보는 아래와 같습니다.

Trigger

Trigger 는 대상이 되는 작업을 실행하기 위한 조건과 실행 방법을 정의한 정보의 집합입니다. Trigger 는 실행할 대상 작업과 작업을 실행하기 위한 조건을 정의하여 생성할 수 있습니다. 작업은 MMC 에 의해 등록된 Job 이어야 하며 작업조건은 시간 기반 방법과 이벤트 기반의 두 가지 방식으로 정의할 수 있습니다.

Trigger 를 정의하는 요소는 다음과 같습니다.

Trigger 기본정보 및 작업수행 조건(Schedule Tab)

작업 실행 조건(Settings): 기본적으로는 시간 기반으로 특정 시점을 기준으로 일정 주기를 설정하여 실행하도록 조건을 지정할 수 있으며, 다른 Trigger 의 실행과 연계하여 특정 Trigger 가 시작되는 시점, 종료되는 시점, Trigger 의 결과값이 True 인 경우에 Trigger 를 실행 하도록 설정할 수 있습니다.

추가 실행 조건(Advanced Settings): Trigger 의 우선순위, 일정시간 이상 작업 시 종료 등의 추가적인 Trigger 실행 조건을 설정할 수 있습니다.

Target Job

Target Job : 실행 대상 작업으로 MMC 에서 등록한 Job 만 대상으로 합니다.

Job Parameter : 실행 대상 작업의 수행 방식을 결정하는 Parameters 값을 설정합니다. Job 에 이미 값이 설정되어 있으나, Trigger 별로 이를 변경해서 실행하고자 할 경우 사용할 수 있 습니다.

Failure Action

Failure Job : Target Job 의 수행 실패 시 수행할 Job 입니다. Target Job 과 같이 MMC 에서 등록한 Job 만 대상으로 합니다.

Job Parameter : Failure Job의 수행 방식을 결정하는 Parameters 값을 설정합니다. Job 에 이미 값이 설정되어 있으나, Trigger 별로 이를 변경해서 실행하고자 할 경우 사용할 수 있습 니다.

Trigger 의 등록/삭제 등의 실제 사용방법은 Trigger 관리방법을 참조합니다.

Shortcut

Shortcut 은 MMC가 Server 와의 파일 교환을 하는 매개체 역할을 합니다. MMC 는 MOZART Server 를 관리하는 도구로서 원격 Server의 특정 Directory 및 File 을 추가, 수정, 삭제할 수 있 어야 합니다. 그러나 보안과 관리 문제로 Server 의 모든 폴더를 접근할 수는 없으며 MOZART 운영/관리 담당자는 MMC를 통해 MOZART Server Working Directory 의 하위 경로만 참조할 수 있도록 구성되어 있습니다. 아래 그림은 Shortcut의 개념을 나타냅니다.



이때 사용되는 개념이 Shortcut으로 MOZART Server 운영을 담당하는 사람은 MMC 의 Server Explorer 에 Server 별 주요 관리 Folder 를 Mapping 한 Shortcut 을 등록하고 Shortcut 을 통해 Server 에 파일 업로드 및 조회작업을 수행할 수 있습니다.

Shortcut 의 사용방법은 Shortcut 관리방법을 참조하십시오.

Project 단위 관리 및 예약 배포 개요

제조 공장 Process 및 정책은 최초 설립이 되고 부터 일관성 있게 흐르는 경우가 없고 시간이 흐름에 따라 변합니다. 이러한 변화에 Server에 등록된 Task DLL 및 Vmodel 파일도 변경 사 항 반영을 위해 주기적으로 업데이트가 필요합니다. Server가 작업을 수행 시에는 이러한 파 일들을 참조를 하고 있기 때문에 엔진 구동 중에 임의로 파일을 교체하게 되는 경우 예상치 못한 오류들이 발생 할 수 있습니다. 비록 Job/Trigger가 예정된 시간 또는 이벤트에 따라 구 동이 되고 종료 시간이 예상이 되어도 사람이 수동적으로 이를 모니터링하여 파일 배포를 하 기에는 한계가 있고 인위적인 실수가 발생하기 마련입니다. 또한 Server의 입장에서는 사용 자가 지정한 WorkingDirectory에 등록된 DLL 및 Vmodel를 참조하게 되어 있어 과거 이력들 에 대해서도 관리자가 신경을 써서 버전 별 관리를 하지 않는 이상 이력 관리가 어렵고 긴급 한 상황에서 Rollback과 같은 대응에도 어려움을 겪을 수 있습니다.

MOZART Management Console (2.0) Client 및 Server 제품에는 위에서 명시된 애로 사항을 완화하고자 사용자가 Project 단위로 파일들을 관리를 하고 Job/Trigger 수행 여부를 체크하 여 파일 배포를 예약 할 수 있는 DeployManagement Service가 제공됩니다. 다음은 Project 관리 및 예약 배포의 개요와 구동 방식에 대해서 기술합니다.

• Project 단위 관리 동작 방식

Project는 Server에서 작업을 수행하는 Job/Trigger가 매핑이 되어 Job/Trigger에서 필 요한 Task의 DLL 및 Vmodel 파일, 엔진 수행 로그 및 결과 파일들을 용도에 따라 관리하 는 관리 요소입니다. Project는 Server의 DeployManagement Service를 통해 동작이 되 며 Project 관련 정보는 DeployManagement DB 파일에 저장이됩니다. DB에 정보를 저장 함으로써 Project를 형상관리가 가능하여 Project 내에 변경 사항이 발생을 하여도 관리 자가 이를 수동적으로 버전 관리를 하지 않아도 됩니다. 아래는 Projects를 통해 파일 배 포하는 동작방식입니다.



MMC를 통하여 파일을 배포 시 해당 파일들은 Project 폴더에 바로 업로드 되지 않습니다. 사 용자가 배포 할 파일들에 대해 배포 일정을 설정을 하고 Commit을 하게 될 시 Changeset 정 보 및 폴더가 생성이 됩니다. Changeset 정보는 DeployManagement.db 파일에 저장이 되고 Changeset 폴더에는 배포 예정인 파일들이 업로드가 되어 배포 시간까지 대기를 합니다. 이 때, DeployManagement Service와 JobScheduler간 통신을 하여 수행 중인 Trigger를 체크를 하고 배포 대상인 Trigger가 수행 종료가 되었을 시 Changeset에 저장된 파일들을 Projects 폴더에 업로드를 하게 됩니다. Changeset 폴더는 변경 사항이 생길 때 마다 생성이 되며 번호 로 관리가 됩니다.



• 예약 배포 동작 원리

하나의 Job에는 다수의 Trigger가 존재 할 수 있습니다. 만약 대상 Job에 DLL 및 Vmodel 파일 배포가 필요 시 사람이 Job에 매핑된 Trigger들의 종료 시간을 다 고려하면서 파일 배포하기란 어려움이 많습니다. 아울러 모든 Trigger가 종료됨을 알았어도 파일 배포 도 중 Trigger가 예약 된 시간에 수행 되는 경우가 있어 이로 인하여 예기치 못한 사고가 발 생 할 수 있습니다. MMC2에서는 DeployManagement 및 JobScheduler Service간 통신 을 하여 DeployManagement에서 배포 대상 파일의 목록을 JobScheduler에게 넘기며 JobScheduler는 목록을 확인하여 대상 파일들을 사용하는 Job/Trigger의 목록을 반환합 니다. DeployManagement에서는 JobScheduler에게 파일 배포를 위하여 대상 Trigger 수 행 제한을 요청합니다. 즉, 배포하는 과정에서 Trigger가 예정된 시각에 시작이 되어 배포 실패를 방지하기 위함입니다. JobScheduler는 대상 Trigger들의 수행을 제한을 하고 DeployManagement에서 대상 Trigger들이 수행되지 않는 것이 확인이 되면 배포 대기 중인 파일들을 Changeset에서 Projects 폴더로 업로드를 합니다. 이후 파일 배포가 완료 가 되면 DeployManagement는 JobScheduler에게 대상 Trigger 수행 제한 해제를 요청합 니다.



Server & Client Settings

개요

MOZART Server 에 의해 구동되는 서비스는 크게 3가지로 Server Setting 을 통해 변경이 가 능합니다.

- Job Scheduler Service : Server 에 등록된 Trigger 정보에 따라 Job 을 실행시켜주는 Server 의 Main Service 입니다.
- **DeployAgent Service** : Server 와의 파일 교환을 위한 서비스로 Upload/Download 기능 을 제공합니다.
- **OutFile Service** : MOZART Job 수행 결과로 Server 에서 압축한 파일의 리스트를 MOZART Studio 에 제공하고 Download 받을 수 있도록 해주는 서비스입니다.
- Licence Service : MOZART Server 에서 일반 사용자들이 사용하는 MOZART Studio 에 대해 License 를 자동으로 발행해주는 서비스 입니다. 자세한 사항은 Local License Service 방식을 참고하시기 바랍니다.
- **DeployManagement Service** : MOZART Server의 Job, Trigger, 파일 배포, Project 등의 변경점 관리를 제공하는 서비스입니다.
- **TriggerJob Service** : Mozart Management Console외에 Web Service와 같이 외부에서 Trigger를 수행 할 수 있도록 제공되는 서비스입니다.

MOZART Server 설치 시 상기 Service 가 모두 실행되며 Service Instance 는 Job Scheduler 만 단독 Instance 로 실행이 되며, 그 외의 Service 는 단일 Interface 로 실행됩니다. 각 Service 별 Server 에서 설정과 Client 에서의 설정 방식은 각 내용 페이지를 참조하시기 바랍니다.

1. Model Download 설정 : Server 에서 Model Download 를 위한 설정 방식과 Client(MOZART Studio) 에서의 설정 방식을 설명합니다.

2. AutoUpdate 설정 : Client (MOZART Studio - 사이트의 구매 제품별 Studio) 에 대한 Update 방법을 설명합니다.

Server 설치는 Install Mozart Server 매뉴얼을 참고하시기 바랍니다.

Model 다운로드 설정

MOZART Studio 에서 Server 에서 수행된 Job 의 결과 파일을 조회하고 다운받아 활용하기 위한 설정 방식을 설명합니다.



Server 설정

MOZART Server 에서 Model Download 설정을 하기 위해 기본적으로 압축된 Model 을 저장 하고 있는 폴더의 **Root**를 지정하고, Client 에서 해당 폴더의 **Model 삭제 여부 및 삭제 시 암** 호를 설정하도록 합니다.

• Server 의 Download Model 파일이 있는 Root Folder 를 지정 : [지정방법] Server 가 설치 된 폴더에서 MozartServiceHost.exe.config 파일 편집을 통해 다음 작업을 수행

值 1.Job Scheduler 에 의해 자동으로 수행된 Model 의 저장 Root 설정

Config Section : <\appSettings>

Key : app-output-dir

설정예시 :



2.개발자, 운영자에 의해 수동으로 수행된 Model 의 저장 Root 설정

Config Section : <\appSettings>

Key : web-output-dir

설정예시 :

1 <appSettings>

- 2 <add key="web-output-dir" value="D:\MozartServer\ModelsManual"/>
- 3 </appSettings>
- Server 의 생성 Model 에 대해 Client 에서 삭제 가능 여부 및 삭제 시 암호 설정 MOZART Studio 상단 메뉴 중 [File > Download Data From Server] 메뉴를 통해 Server 의 압축된 Model 을 조회할 때 Model 에 대한 삭제를 할 수 있는데, 이때 삭제하기 위한 암호를 설정합니다. 설정 방식은 Download Root Folder 설정방식과 동일하게 Config 를 편집합니다. Setting 을 위한 Key 와 Section 은 아래와 같습니다. Config Section : Key : password

설정예시 :



※"password" 를 설정하지 않는 경우 Client 에서 압축된 Model 을 삭제할 수 없습니다.

Client 설정

Client 에서는 Model Download Server 와 해당 Server 에서 Model 별 저장 폴더를 지정해주어 야 합니다. 설정 방식은 각 사이트에서 구매한 **OOO_Studio** 를 실행하여 [**Tool > Options**] 메뉴 를 통해 설정합니다.

M Preferences				×
▲ (General) AutoUpdater	Output download	d sites		+ - 1 +
Downloads	Name	SubDir	Url	
	TestModel	TestModel	http://192.168.1.46:8000/mozart/OutFile	Service
	Test2	Test2Model	http://192.168.1.46:8000/mozart/OutFile	Service
			ОК	Cancel

- 왼쪽 Tree 에 Downloads 를 선택합니다.
- 오른쪽 리스트에 Download 사이트를 추가합니다. 추가할 때는 상단의 [+] 버튼을 이용합 니다.

🖳 Add site	
Name	Test2
SubDir	Test2Model
URL	http://192.168.1.46:8000/mozart/OutFileService
	ex) http://192.168.25.61:8080/mozart/OutFileService
	OK Cancel

☑ Name : 추후 Model Download 시 콤보박스에 표시되는 사이트 명입니다.

- SubDir : Server 에 Model 이 저장되어있는 폴더 명입니다. Server 에서 설정한 Model Download Base Folder 에서부터 상대경로로 지정됩니다.
- URL: Model 파일 Download 서비스를 제공하는 Server 의 서비스 URL 입니다. 형식은 위의 예시와 같은 방식으로 설정하며, 사용자 사이트의 운영팀을 통해 확인한 URL 을 입력합니다. 일반적으로 Port 설정과 IP 가 다를 수 있습니다. 이는 Server Setup 후 운영팀을통해 확인합니다.
- 여러 개의 사이트를 등록할 수 있으며 상단의 화살표로 Download 창에서 표시순서를 조 정할 수 있습니다.
- 등록된 사이트의 내용을 수정하고자 할때는 더블 클릭하거나, 상단의 [...] 버튼을 사용합니다.
- 추가한 대상을 삭제하고자 하는 경우에는 [-] 버튼을 사용합니다.

상기 예시와 같이 하나의 Server 에 여러 개의 폴더를 나누어서 등록하거나 여러 개의 Server 를 등록 할 수 있습니다. 위와 같이 설정한 후 Studio 에서 **[File > Download Data From** Server] 메뉴를 통해 Model 검색 창을 띄워 Model 을 Download 할 수 있습니다.

	File	Edit	Project	Data	Tool	Windov			
٦	6	New I	Model		Ctrl+	N			
	7	Open	Project		Ctrl+	-0			
		Close							
		Close	Project			- 1			
-		Save			Ctrl	+S			
!	ø	Save	All	Ctrl	+Shift-	+S			
	Ъ.	Print			Ctrl	+P			
	Ø	Print	Preview			- 1			
;		Print Settings							
		New Target							
•		Open Tareget							
		Change Target/Parent							
		Link Models							
		Recen	it File List			- - -			
		Down	load Data	a From	Server	·			
		Exit							

메뉴를 실행하면 아래 그림과 같이 Download Server 콤보박스에 등록한 Name 별로 리스트 가 뜨고 해당 리스트를 선택하는 경우 해당 폴더에서 Download 가능한 Model 의 리스트를 조회하여 Download 할 수 있습니다.

AutoUpdate 설정

Auto Update 는 사용자 사이트에서 직접 개발한 UI Component 외에 Config 파일과 같은 설 정파일을 업데이트하는 경우에 사용됩니다. 즉, 사용자 조직의 특정 Server 에 업데이트된 파 일을 압축된 형태로 올려놓으면, 사용자 PC 에서 초기에 설정한 Server 로 연결하여 업데이트 파일이 있는 경우 이를 자동으로 업데이트할 수 있습니다. 구조는 아래 그림과 같습니다.



이러한 설정을 하기 위해서는 IIS 가 설치된 회사 내 Server 를 Update Server 로 사용하거나 Auto Update 용도로 별도로 Server 설정을 해주어야 합니다. 아래는 Server 설정 방법입니다.

Server 의 사전 준비사항

.NET Framework 4.0 이상 IIS (Internet Information Server) version 6 이상

Server 설정 방법

1. 업데이트 대상 파일을 저장해둘 폴더를 정합니다.

2. 'IIS (인터넷 정보 서비스) 관리자'를 실행합니다.

3. [응용 프로그램 풀 > 응용 프로그램 풀 추가] 메뉴로 응용 프로그램 풀을 추가합니다. 이때 .NET 버전을 4.0으로 설정합니다. (응용 프로그램 명은 임의로 정의합니다. 예: MozartUpdateServer)

4. [사이트 > Default Web Site > 응용 프로그램 추가] 메뉴를 사용하여 응용 프로그램을 추가 합니다. 아래와 같이 입력값을 설정합니다.

- 별칭: 등록할 응용 프로그램의 별칭을 입력합니다. 별칭은 Client 에서 Server URI 입력 시 필요한 명칭입니다. Client 에서 Download URL 값 설정 시 "host 주소/[별 칭]/manifests.xml" 과 같이 입력해야 합니다.
- 응용 프로그램 풀: [선택] 버튼으로 2번에서 등록한 응용 프로그램 풀을 선택하여 입력합 니다.
- 실제 경로: 1번에서 설정한 업데이트 대상 파일을 저장해놓을 폴더를 지정합니다.

5. MOZART Client 배포 시 설치된 Auto Updater 폴더에 ManifestEditor 를 사용하여 Manifest 파일을 편집합니다. 자세한 내용은 Manifest 파일 편집방법을 참고하시기 바랍니다.

6. 신규 생성 혹은 작성한 Manifest.xml 파일을 업데이트 대상 폴더에 복사하면 Server 설정 은 완료됩니다.

♀ Default Web Site 에서 사용하는 Port 가 열려있어야 합니다. 기본적으로 사용하는 80
 Port 의 경우에도 Server 에서 막아 놓은 경우 오류가 발생할 수 있으니 Server 설정 시 확인하
 시기 바랍니다.

Client 설정 방법

아래와 같은 순서로 MOZART Studio 에서 클라이언트 UI 및 관련 파일에 대한 자동 업데이트 설정을 할 수 있습니다.

1. Client 에서는 Studio 의 [Tool > Options] 메뉴를 사용하여 Auto Update 여부 및 업데이트 Server 를 설정합니다.

2. 각 설정항목은 아래와 같이 설정합니다.

🗃 Preferences		×
▲ (General) AutoUpdater		✓ Auto update
Downloads	Application Id	{3A8F794F-D23A-484D-B766-98581D801DFD}
		ex) {3A8F794F-D23A-484D-B766-98581D801DFD}
	Download URL	http://192.168.1.46/studioupdate/manifests.xml
		ex) http://192.168.255.255/Update/manifests.xml
	Downloader	BITSDownloader Use this downloader
	Parameters	userName= password= authenticationScheme=BG_AUTH_SCHEME_NTLM targetServerType=BG_AUTH_TARGET_SERVER
		OK Cancel

- Auto update : 체크를 하면 하부에 입력된 정보를 기준으로 자동 업데이트가 되며, 체크 해제 시 자동 업데이트를 하지 않습니다.
- Application Id : Studio 프로그램의 고유 아이디입니다. 사용자가 변경하지 않습니다. Server 의 Manifest 파일과 비교할 때 대상 Application 에 대한 업데이트 정보만 비교합 니다.
- Download URL : Update Server 의 URL 을 입력합니다. 형식은 다음과 같습니다.
 + 형식 : http://[Server IP]/[Server 설정 시 응용 프로그램 별칭]/manifests.xml
 + 변경부분은 [Server IP], [Server 설정 시 응용 프로그램 별칭] 입니다. Client UI 개발/운 영조직에게 해당 부분을 확인하고 설정합니다.
- Downloader : Downloader 를 선택합니다. 기본은 BITDownloader 가 선택되어 있습니다.
- Parameters : Server 와의 인증에 사용되는 Parameters 로 수정하지 않습니다.
- 3. 항목 설정 후 Studio를 재시작하면 아래와 같은 다운로드 선택창이 활성화됩니다.

🎄 Software Up	odate	×
	A new version of FP_StudioUpdate is available! FP_StudioUpdate 1.0 is now available. Would you like to download it now?	
	Releae Notes:	
	FP_studio update test	
	👉 Skip this version 🔯 Remind me later 💿 Update	

- Skip this version : 다음번 실행 시에도 해당 버전의 업데이트를 체크하지 않습니다.
- Remind me later : 다음번 실행 시에 해당 버전의 업데이트 수행 여부를 묻습니다.
- Update : 업데이트 내용을 다운로드 받고 Studio 를 재실행합니다.

Manifest Editor 사용법

Manifest Editor 는 Manifest 파일을 생성/편집하기 위한 편집기입니다. MOZART Client 설치 시 MOZART 설치경로 아래의 Updater 폴더에 ManifestEditor 가 함께 설치됩니다.

1. Server 실행파일 폴더의 ManifestEditor 를 실행합니다. 총 4개의 Tab 으로 구성된 다이얼 로그가 실행됩니다.

	Downloader	Application Properties	Activation Process		
Manifest Propertie	c		1		
	-				
ManifestId				Generate	Mandatory
Title				Version	
Release Notes					×.
ncluded Manifests Location					
Manifest Id					
		Add Remove			

2. Manifest Properties Tab 의 내용을 입력합니다.

- Manifestld : 신규 업데이트 파일이 배포될 때마다 변경되는 배포 ID 와 같습니다. Client 는 해당 Manifestld 의 값을 비교하여 Auto Update 여부를 판별합니다. 우측의 [Generate] 버튼을 통해 신규 GUID 를 생성할 수 있습니다. 배포 시마다 변경되어야 합니다. (※xml 문서 Key = manifestld)
- Title : 해당 Manifest 의 이름입니다. (※xml 문서 Key = title)
- Version : 배포되는 제품의 버전을 기록합니다. 배포 시 규칙에 따라 업데이트합니다. (※xml 문서 Key = version)
- Release Note : 배포되는 버전의 수정사항에 대해 간략히 기록합니다. 배포 내용에 따라 업데이트합니다. (※xml 문서 Key = description)

(2) was a second to a second t
<pre></pre>
<pre><manifest pre="" xmlns:xs<="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></manifest></pre>
<pre>manifestId="{AF7A3BD5-10A1-4155-BBF8-631906D86DAE}" mandatory="False" xm</pre>
<title>FP Studio</title>
<version>1.0.3</version>
<description>first release</description>
<pre><application applicationid="{3A8F794F-D23A-484D-B766-98581D801DFD}"></application></pre>
<pre><entrypoint file="FP_Studio.exe" parameters=""></entrypoint></pre>
<location>.</location>
<files <pre="">base="http://xxx.xxx.xxx/MozartUpdate" hashComparison="No"></files>
<file source="update-files.zip" transient="No"></file>
<activation></activation>
<tasks></tasks>
<task mozart.autoupdater.activationprocessors.applicationdep<="" pre="" type="Mozart.AutoUpdater.ActivationProcessors.WaitForApplica</th></tr><tr><th><task <pre>type="></task>

상기 예시 파일은 **[Mozart Client 설치 폴더 > Updater > Server]** 에 manifests.xml 파일로 포 함되어 있습니다.

3. Application Properties Tab 의 내용을 입력합니다.

업데이트 대상이 되는 Application 의 정보를 설정합니다. 필수 설정 항목은 아래와 같습니다.

📝 Updater Manifest Too		
Manifest Properties	Jownloader Application Properties Activation Process	
Application Properti	es Entry Point	
Application Id	File	
Location	Parameters	
Files		
Files URI	Use hashing md5 🔹	
Source Folder	E:\Temp\	Browse
Files		
		Add
New	Open Save Validate View	Close

- Application Properties
 - ApplicationId : 업데이트 주 대상 프로그램의 GUID 로 대상 파일의 App.config 파일
 에 있는 GUID 를 사용합니다. 두 개의 GUID 를 항상 같도록 설정하면 됩니다. (※xml 문서 Key = application/applicationId)
 - Location : Download 되는 파일 저장 위치입니다. (※xml 문서 Key = application/location)
- Entry Point
 - File : 실행 파일 명입니다. FP_Studio 에 대한 업데이트를 설정하는 경우
 'FP_Studio.exe' 로 설정합니다. (※xml 문서 Key = application/entrypoint file)
 - Parameters : 실행 시 Parameters 를 설정합니다. Studio 업데이트 시 별도 Parameters가 필요하지 않습니다.
- Files
 - Files URI : Download 받을 파일이 위치한 URI 를 설정합니다. 일반적으로 Server 에 서는 LocalHost 에 설정한 경로를 입력합니다. (※xml 문서 Key = files/base)
 - Source Folder : 파일이 위치한 로컬 폴더를 선택합니다.
 - Files : 선택한 로컬 폴더의 파일 중 Download 받을 파일 명을 입력합니다. 일반적으 로 업데이트할 파일을 압축한 파일로 선택합니다. (※xml 문서 Key = files/source)
- 4. Activation Process Tab 의 내용을 입력합니다.

Update 시 실행할 Processor 를 선택합니다. 복수로 선택 가능합니다.

🕞 Updater Manifest Tool	
Manifest Properties Downloader Application	Properties Activation Process
Processor Name	
Processor Type Application Deploy	▼ Add
Processors List ApplicatoinDeployProcessor - Mozart.AutoU	Jpdater. ActivationProcessors. ApplicationDeployProcessor, Mozart. AutoUpdater, Versio
WaitForApplicationExitProcessor - Mozart.Au	utoUpdater.ActivationProcessors.WaitForApplicationExitProcessor, Mozart.AutoUpdate
	Remove
<	۲ (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰۰۰) (۲۰
New Open	Save Validate View Close
Some files could not be added because they aren't	under the specified Source Folder.

Processor Name 을 입력하고 Processor Type 선택 후 [Add] 버튼으로 Processor 를 추가합니 다. Processor Type 의 유형은 아래와 같습니다. 여러 Process 를 함께 등록할 수 있습니다. MOZART Studio 의 자동 업데이트를 위해 꼭 필요한 항목은 Application Deploy, Wait For Exit 두 가지입니다. Server 설치 폴더에 함께 배포된 manifests.xml 파일을 참고하시기 바랍 니다. (※xml 문서 Key = activation/tasks/task)

- Application Deploy : 다운 받은 파일을 업데이트 대상 폴더로 복사, 파일이 압축 파일이 면 압축 해제합니다.
- File Copy : config 파일에 설정된 대로 특정 파일을 특정 위치로 복사합니다.
- File Delete : config 파일에 설정된 대로 특정 경로의 특정 형식의 파일들을 삭제합니다.
- Folder Copy : config 파일에 설정된 대로 특정 폴더를 특정 위치로 복사합니다.
- Folder Delete : config 파일에 설정된 대로 특정 폴더를 삭제합니다.
- **GAC Util** : GACUtil을 사용해서 config 파일에 설정된 대로 GAC 관리합니다. (등록/삭 제/...)
- Hash Validation : 다운 받은 파일 중 config에 설정된 특정 파일과 Server 의 원본 파일 간 Hash 값 비교합니다.
- Install Util : InstallUtil을 사용해서 Service 를 관리합니다.
- MSI: config 파일에 설정된 패키지를 설치/제거/패치합니다.
- Start Application : 파일 다운로드 및 업데이트 완료후 프로그램을 다시 시작합니다.
- Uncompress : config 파일에 설정된 대로 특정 압축 파일을 특정 경로에 압축 해제합니다.

• Wait For Exit : 업데이트를 위해 Client 를 종료하고, 업데이트가 완료되기를 기다립니다.

MMC User Guide

서버 관리

MMC 를 통해 복수 개의 Server 에서 작업관리가 가능합니다. MMC 의 관리 대상 Server 를 등 록하기 위해서는 해당 Server 에 MOZART Server 가 설치되어 있어야 합니다. Server Explorer 의 **[Add Server Connection]** 버튼을 사용하여 관리 대상 MOZART Server 를 추가 할 수 있습 니다.

Server 등록 방법

- 1. Server Explorer 의 [Add Server Connection] 버튼(🚯)을 클릭합니다.
- 2. 입력 다이얼로그의 입력항목을 입력합니다.

Edit && Connect Computer	<
Input displaying unique name:	
192, 168, 1, 10	
Input the computer you want to manage:	
Server: http://192,168,1,10 Port: 8000 App Name: mozart]
Input login infomation:	
∗ User ID 🛛 sa 🕢 🗹 Remember User ID	
* Password	
Ok Cancel]

Input name : Console 관리자가 대상 Server 를 관리하기 위한 이름을 입력합니다. **Input the computer you want to manage** : 서버 접속에 필요한 정보를 입력합니다.

- Server : 해당 Server 의 URL 을 입력합니다. 입력 예시에 표시된 형태로 입력하되, 실제 Server 의 IP 주소를 바르게 입력합니다.
- Port : 대상 Server의 Port 번호를 입력합니다. Default는 8000번이며, MozartSericeHost.exe.conig 및 MozartJobScheuler.config에서 Port 번호를 변경을 하였 을 시 해당 Port 번호를 입력합니다.
- App Name : 대상 Server의 Application Name을 입력합니다. Default는 "mozart" 입니다.
User ID : MOZART Server에 접속 할 사용자ID를 입력합니다. (최초 설치 시 Default 관리자 계정은 sa 입니다.) Password : MOZART Server에 접속에 필요한 비밀번호를 입력합니다. (sa 계정의 초기 비밀 번호는 "mozart" 입니다.)

MOZART Server 최초 설치 될 시 Deployment.db 파일 내에 기본 관리자 계정 및 비밀번호가 포함되어 있습니다. 처음 접속 시에는 기본 접근 정보로 접근 후 이후 사용자를 추가하게 될 시 사용자 별 접근이 가능하게 됩니다. 최초 설치 시 기본 관리자 계정/비밀번호는 다음과 같 습니다.

- User ID : sa
- Password : mozart

3. 모두 입력한 후 [OK] 버튼을 클릭합니다.

4. Server Explorer 에 입력한 이름으로 Server가 생성되면 서버 등록이 끝납니다.



서버 상태 확인

MMC Server Explorer에 등록된 Server의 상세한 정보를 확인 할 수 있습니다. Server 이름 및 IP주소와 같은 기본 정보부터 Server 머신의 H/W 사양, Server에 설치된 MOZART DII 버전, WorkingDirectory 및 Backup으로 사용하고 HDD 용량 상태 등도 확인이 가능합니다. 아래는 MMC Server Explorer에서 Server 정보를 조회하는 방법과, 각 항목의 정의를 기술합니다.

Server 정보 조회하기

1. Server Explorer에서 정보 조회를 하고자 하는 대상 Server에 오른쪽 마우스 클릭을 하여 [Server Information] 을 선택합니다.



2. Server Information Dialog에서 Server 정보를 확인합니다.

🔢 Server Information 🛛 💽									
Connection									
Server Name:		MMC2_TES	MC2_TEST						
Server Addre:	\$\$:	http://loca	lhost:8000/mozart	:					
Server Speci	fication	1							
Processor:		Intel(R) Cor	re(TM) i7 CPU	960 @ 3.20G	Hz				
Total Memor	y:	8 GB							
OS:		Microsoft \	Vindows 7 Profess	ional K (64bit	t)				
.NET Framew	ork:	4.0.30319.1							
– Installed Com	npone	nt							_
Name			Creation Time		File	e Version			
MozartService	Host.e	exe	2017-03-28		20)17.1.108.0			
Mozart.Comm	on.dll		2017-03-28		2017.1.108.0				
Mozart.Compr	ession	.dll	2017-03-28		20	2017.1.108.0			
Mozart.Core.d			2017-03-28		20	2017.1.108.0			
Mozart.Data.d			2017-03-28		20)17.1.108.0			
Mozart.Data.E	ntity.c		2017-03-28		20)17.1.108.0			
Mozart.Data.T	'ools.d		2017-03-28		20)17.1.108.0			-
– Disk –									
Disk Name	Usage	∋s		Using Ratio		Total (GB)	Remair	n (GB)	
E:₩	Work	ing Director	y, DataBackup	<mark>21</mark> .82 %		55			43
C:₩	FabCo	opyBackup		51.39 %		72			35
Performance									
Trend Refresh									
Item Current Average Usage Ratio									
CPU 12.26		12.26 %		6.64 %					
Memory		18.83 %(1.51 GB)		18.44 %					
Trigger Count			0						

Connection : 대상 Server의 접속 정보를 확인하는 영역입니다.

- Server Name : Server Explorer에 Server 등록 시 입력한 서버 이름 정보입니다.
- Server Address : Server Explorer에 Server 등록 시 입력한 IP 주소 정보입니다.
 - Server Specification : 등록된 Server의 H/W 사양, OS 및 .NET Framework 버전 정보 를 표시합니다.
 - Installed Component : Installed Component에서는 대상 서버에 설치된 MOZART의 DII 버전을 확인 할 수 있습니다.
 - Disk : 대상 Server의 HDD 용량 정보를 표시 합니다. Disk에는 WorkingDirectory와 Backup이 위치한 HDD에 대한 정보만 표시합니다. (표시 단위 : GB)
 - **Performance**: 대상 Server의 현시점의 CPU 및 Memory의 사용량, 수행 중인 Trigger 의 개수를 표시합니다.

- Trend : 현재 시점의 사용량 외에 대상 서버의 CPU/Memory 사용 현황을 확인 할 수 있습니다. Trend 버튼 클릭 시 Performance Trend Dialog가 팝업됩니다.
- **Refresh** : Performance 정보를 가장 최근 정보로 새로고침합니다.

Performance Trend 보는 방법

Server Information Dialog 하단 Performance 영역에서 Trend를 클릭하면 Performance Trend Dialog가 열립니다.

Performance Trend Dialog를 통해 현재를 기준으로 사용자가 지정한 기간 동안의 전반적인 Server의 자원 사용 및 Trigger 실행 현황 확인이 가능합니다.



아래는 MMC의 Performance Trend Dialog의 화면입니다.

- Period : 기간을 설정합니다. 기간은 시간/일 단위로 설정 할 수 있으며 현재 시점을 기준 으로 입력된 숫자 기간 전 만큼의 Server 현황이 그려집니다.
 - Average : 사용자가 지정한 현재로부터 입력된 기간 전까지의 평균 CPU/Memory 사용량 및 Trigger Count를 표시합니다.

사용자가 Period를 설정하면 Server 현황이 그래프로 표시됩니다. 그래프의 점선은 해당 기 간 동안 평균 CPU/Memory 사용량 뜻하며, 실선은 해당 시간의 실제 CPU/Memory 사용량을 나타냅니다.

또한 그래프에 마우스 오버를 할 시 해당 기간에 대한 상세 정보를 확인 할 수 있는 팝업이 나 옵니다.

Local DB 파일 다운로드 방법

Project 생성 및 변경 이력, Job 이력, Trigger 변경, 실행 이력 및 사용자 정보 등의 정보들은 Server의 Local DB (SQLite DB 파일)에 기록이 됩니다. 네트워크 문제 등 기타 장애로 인하여 DB 기록 누락 사항이 발생 할 수 있습니다. 이러한 경우 LocalDB 파일로 부터 정보를 제대로 가져오지 못하여 등록된 Job/Trigger가 갑자기 안 보이거나 Project의 등록된 파일들을 읽어 오지 못하는 문제 등이 발생하는 경우가 있습니다. 문제를 해결하기 위해서는 문제가 되는 Row를 처리해야 하며, 그러한 작업 진행을 위해 LocalDB 파일을 다운받아 제품 관리자에게 전달을 해야 합니다. 다음은 MMC를 통해 LocalDB 파일을 다운로드/업로드 방법에 대해서 기술합니다.

LocalDB 파일 다운로드 방법

1. LocalDB 파일을 다운르도 받을 대상 서버 노드를 Server Explorer에서 선택합니다.

2. 선택한 서버 노드에 마우스 오른쪽 버튼 메뉴를 통해 [Download Database File] 를 선택합니다.



3. LocalDB 파일을 다운로드 받을 경로를 지정하고 [확인] 버튼을 클릭합니다.

폴더 찾아보기	×
Select a Download Folder	
> 📑 동영상	^
> 🔮 문서	
> 🔜 바탕 화면	
> 📰 사진	
> 👌 음악	
✓ 🏪 로컬 디스크 (C:)	
DB_File_Download	
GridSettings	
> Logs	
> ODAC	
> ODAC112040Xcopy_32bit	~
새 폴더 만들기(<u>M</u>) 확인 취소	

4. 다운로드가 진행이 되면 아래와 같이 Progress Bar가 나옵니다. 다운로드 중 취소를 하고 싶으면 **[Cancel]** 버튼을 클릭합니다.

Transfer files X	
Total files (0/1)	
C:\Windows\ServiceProfiles\LocalService\AppData\Roaming\Mozart\DeployMa	n
<u>C</u> ancel	

5. LocalDB 파일 다운로드가 완료되면, "Done"이라는 팝업 메시지가 출력됩니다. [확인] 을 클 릭합니다.

×	
Done!	
확인	

6. 다운로드 경로에 **"DeployManagement.db"** 라는 파일이 존재하면 LocalDB 파일을 정상적 으로 파일을 다운로드 한 것입니다.

LocalDB 파일 업로드 방법

1. LocalDB 파일을 업로드 할 대상 서버 노드를 Server Explorer에서 선택합니다.

2. 선택한 서버 노드에 마우스 오른쪽 버튼 메뉴를 통해 **[Upload Database File]** 를 선택합니다.



3. 업로드 할 DeployManagement.db 파일의 위치를 지정합니다.

🥖 열기					×
← → × ↑ 📙 → 내 PC → 로컬 디스	노크 (C:) → DB_File_Download		✓ [™]	DB_File_Download 검색	Q,
구성 ▼ 새 폴더					?
images ^	이름	수정한 날짜	유형	크기	
MozartManual	NeployManagement.db	2018-05-04 오후	Data Base File	158,033KB	
·····································					
옴페이지 중독 이미지					
a OneDrive					
💻 내 PC					
3D 개체					
➡ 다운로드					
등 동영상					
문서					
바탕 화면					
▶ 20					
♪ 음악 ··· 르러 티스크 (C)					
파일 이름(N): Deploy	/Management.db		~	Mozart Server Databse Files	(C ~
				열기(<u>0</u>) ▼ 취소	

4. Server 파일이 이미 존재하는 경우 파일을 덮어쓸지 복사본을 만들지 선택을 합니다. [Yes] 를 클릭 시 기존 파일을 덮어씁니다. [Copy And Rename]을 선택 시 파일의 복사본을 만듭니 다.

E Confirm File Replace	Х
The folder already contains a file named 'DeployManagement,db',	
Would you like to replace the existing file Size: 154,33 MB Date modified: 2017-05-30 09:29:40	
with this one? Size: 154,33 MB Date modified: 2018-05-04 14:42:15	
or Copy and Rename as DeployManagement (1),db	
Yes No Copy And Rename Cancel	

Server가 운영되는 동안에는 Server의 LocalDB 파일에는 지속적으로 이력이 기록이 됩니다. 만약 LocalDB 파일을 업로드하기 전 Trigger 실행 이력을 제외한 Projects/Job/Trigger에 파 일이나 설정 변경이 있는 경우에는 [Copy And Rename]으로 파일을 업로드 하는 것을 권장 드립니다.

이후 원본과 복사본 파일을 비교하여 테이블을 합치는 쿼리를 수행하여 원본 DB 파일에 오 류 수정과 최근 기록을 유지하도록 합니다. 또한, 이러한 작업을 진행 시에는 쿼리가 실행되 는 동안 원본 DB파일에 추가적으로 기록되는 것을 피하기 위해 Server Explorer에서 대상 Server를 선택하여 [Pause Job Scheduler]를 실행하여 Job Scheduler Service를 임시 중단합 니다.

5. 업로드가 진행이 되면 아래와 같이 Progress Bar가 나옵니다. 업로드 중 취소를 하고 싶으 면 **[Cancel]** 버튼을 클릭합니다.

Transfer files	\times
Total files (0/1)	
C:\Windows\ServiceProfiles\LocalService\AppData\Roaming\Mozart\Deplo	yMan
<u>C</u> ancel	

6. LocalDB 파일 업로드가 완료되면, "Done"이라는 팝업 메시지가 출력됩니다. **[확인]** 을 클릭 합니다.

×
Done!
확인

Project 관리

Project는 실행 대상 Job/Trigger와 매핑되는 Vmodel 파일과 Task DLL 파일 관리 및 업데이 트 이력을 관리하기 위한 관리 요소입니다. Project를 통해서 Vmodel 및 DLL 파일을 예약 배 포를 할 수 있으며, 형상관리의 개념이 도입되어 배포 예약 시점마다 변경점을 생성하여 이력 관리가 가능합니다. Job/Trigger가 Project와 매핑이 되어 Project에서 변경 사항이 생기면 매 핑된 Job/Trigger의 수행 여부를 체크하여 수행 중이 아닌 경우에 파일을 배포 합니다.

- Project 등록하기 : Project를 등록하는 과정을 설명합니다.
- Project 편집/삭제 : 사용자가 추가한 Project를 삭제하는 방법에 대해 설명합니다.
- File Commit 및 Deploy : Project를 통해 파일 및 폴더를 예약 배포 하는 방법에 대해 설명 합니다.

File Commit 및 Deploy하기

File 등록하기

1. Server Explorer->Projects 노드에서 파일 등록이 필요한 Project의 Files 폴더를 더블 클릭 또는 마우스 오른쪽 버튼 메뉴의 **[Open]** 을 선택합니다.

2. 상단 메뉴의 . [Add] 버튼을 클릭하여 업로드 할 파일 및 폴더를 선택합니다. 새롭게 추가가 되는 경우 파일 옆에 "+" 표시가 되고, 삭제하는 경우는 "-" 그리고 변경되는 경우에는 아래 그림과 같이 체크 표시가 나타납니다.

💽 - 💿 - 🜫 📴 🕞 - 📄 📄 🛄 🍮 🕀 Server Test				
Name	Last Update Time	Description		
🚺 Data	2017-03-27 오후 6:32:30			
Model.vmodel	2017-03-27 오후 6:32:30			
ServerTest.pdb	2017-03-27 오후 6:32:31			
🖌 🖉 ServerTest.dll	2017-03-27 오후 6:32:31			

상단메뉴 버튼 설명

- 💋 : Files 폴더 내의 파일들의 Deploy 상태를 업데이트 하기 위한 Refresh 버튼입니다.
- 💦 : Files 폴더 내에 폴더를 생성하기 위한 Create Folder 버튼입니다.
- _____ : Files 폴더에 파일/폴더를 등록/변경하기 위한 Add 버튼입니다. 아이콘 옆의 화 살표를 누를 시 파일 혹은 폴더 단위로 업로드 할 여부를 선택 가능합니다.
- 📝 : Files 폴더 내에 파일 변경을 위한 Update 버튼입니다. 기능은 Add와 동일하지만 Update로 할 시 선택한 파일만 탐색기에서 검색됩니다.
- 📄 : Files 폴더에서 선택한 파일을 삭제하기 위한 Delete 버튼입니다.
- 📄 : 상기 Create Folder/Add/Update/Delete을 반영하기 위한 Commit 버튼입니다.
- 👌 : Files 폴더 내 모든 변경 사항을 취소하기 위한 Cancel 버튼입니다.

• • Files 폴더 내에 현재부터 과거 모든 변경 사항들을 조회하기 위한 View History 버튼입니다.

	History : Server Test					
	Changeset	Action	User	Deployed Time	Comment	
	32	Deploy	sa	2017-03-27 18:32		
	50	Deploy	sa	2017-04-11 11:47		
	51	Deploy	sa	2017-04-11 12:05		
	52	NotScheduled	sa	0001-01-01 00:00		
l						
	Detail					ОК

View History Window 정보

- Changeset : 변경점 번호를 표시합니다. 변경점 번호는 Project 별로 생성되지 않고 (i.e. Project A : Changeset 1, Project B : Changeset 2,...) 생성된 모든 Project들 중 한 곳에서 라도 Commit이 발생하면 이전 Changeset 번호에서 +1이 됩니다. (i.e. Project A: Changeset 1, Project B: Changeset 2...)
- Action : Commit에 대한 Action 정보를 나타냅니다.
 - Deploy: Commit된 파일들이 Server에 배포가 완료된 상태를 나타냅니다.
 - *NotScheduled* : 파일들이 Commit이 되어 Changeset 폴더에는 저장이 되었지만 아 직 Server에는 배포되지 않은 상태를 나타냅니다.
 - Deploy > Rollback : Job/Trigger가 수행 중이어서 배포/삭제가 가능하지 못한 상태 인 경우 Commit되는 파일이 Rollback 된 상태입니다. 이는 Deploy Schedule에서 Related Items가 선택되지 않은 상태에서 Deploy 시도 시 관련된 Job/Trigger가 수행 중인 경우 발생합니다. Deploy > Rollback이 되는 경우 Comment에 오류 메시지가 남습니다.

- User : Commit을 진행한 사용자 계정을 나타냅니다.
- **Deployed Time**: Commit된 파일들이 Server로 배포된 시간을 나타냅니다. (형식: YYYY-MM-DD hh:mm).
- **Comment** : Commit 할 시 사용자가 남긴 Comment를 표시합니다.

File Commit 및 Deploy하기

1. 파일 등록이 완료가 되었으면 상단 메뉴의 📑 [Commit] 버튼을 클릭합니다. 이때 File들 을 배포할 시점을 Deploy 탭에서 설정 할 수 있습니다.

2. 설정이 완료되면 **[OK]** 버튼을 클릭합니다.

🕺 Commit Changes			
Project: Server Test			By User : sa
Files Deploy Schedule			
Update file			
Name	Directory	Description	
✓ ServerTest.pdb	Server Test		
			OK Cancel

Commit Changes Dialog 정보

• Files Tab

- Comment : 변경점에 대한 사용자 Comment를 작성하는 영역입니다. 작성된
 Comment는 (A) [View History] 를 통해 확인이 가능합니다.
- **Name** : Commit 할 파일 이름 정보입니다.
- Directory: [WorkingDirectory]/Projects/ 경로의 [Project 이름]에 대한 상대 경로 정 보입니다.
- Deploy Schedule Tab

Commit Changes		
Project: Server Test		By User : sa
Files Deploy Schedule		
 Update Now Update after specific time No schedule yet 	Related Type: Job • Related Items: (NONE) • Kill executing triggers	
		OK Cancel

- Update Now : [OK] 버튼을 누르는 시점부터 파일 배포를 시작합니다. Related Items가 지정된 경우라면 관련 Job/Trigger가 수행 종료 후 파일 배포를 합니다. Related Items가 (NONE)인 상태에서 관련 Job/Trigger가 수행 중이면 배포는 취소되고 파일들은 Rollback 됩니다.
- Update after specific time : 사용자가 지정한 날짜 및 시간에 Commit 한 파일들을 Server에 배포합니다. Update Now와 마찬가지로 Related Items가 지정되지 않는 경우 배 포 할 시간에 Job/Trigger가 수행 중이면 Rollback이 됩니다.
- No schedule yet : 등록/변경 파일들을 Commit만 하고 Server에는 아직 배포하지 않을 때 선택하는 옵션입니다. No schedule yet인 파일들은 나중에라도 Update Now 또는 Update after specific time을 통해 Server에 파일들을 배포 할 수 있습니다.

- Related Type : 등록/변경된 파일들을 배포 시 관련된 Job 또는 Trigger 종료 여부를 체크 를 할지 선택하는 항목입니다. Related Type을 Job로 설정 시 Related Items에서 선택된 Job과 연결된 모든 Trigger의 수행 여부를 체크를 하여 배포를 진행을 하고 Trigger로 설 정하는 경우 Related Items에서 선택한 Trigger들의 수행 여부를 체크하여 배포를 진행합 니다.
- **Related Items** : Related Type에 설정한 항목에 따라 등록된 Job 또는 Trigger를 선택하 여 수행 여부를 체크합니다.
- Kill executing triggers : Related Type/Related Items의 선택된 Job/Trigger가 수행 중인 경우 Kill executing triggers가 체크되어 있으면 해당 Job/Trigger의 수행을 즉시 중단하 여 파일들을 배포합니다. 해당 옵션은 파일 배포가 Trigger 수행보다 우선일 경우 선택을 합니다.

Project 등록하기

기존 MOZART Management Console 에서는 Shortcut을 활용하여 Task model 및 dll 업로드, 로그 및 모델 결과 압축 파일 조회를 하였다면, MOZART Management Console (2.0)에서는 Project를 생성 시 자동으로 Project의 Files/Logs/Result 폴더가 생성이 되어 별도의 Shortcut 생성 없이 Project 단위로 관리 및 조회가 가능합니다. 다음은 Project를 등록하는 방법에 대 해서 기술합니다.

Project 등록하기

1. Server Explorer -> Projects 노드에서 마우스 오른쪽 버튼 클릭을 하여 **[Add Project]** 를 선 택합니다.



2. Input project name 및 Description의 내용을 작성하여 [OK] 버튼을 눌러 저장을 합니다.

Add New Project	
Input project name:	
My Project	
Server folder:	
workingFolder\My Project\	
Description:	
새로운 프로젝트 추개	
	OK Cancel

Add New Project 정보

- Input Project name : 생성할 Project의 이름을 작성하는 Text box입니다. WorkingDirectory 내 Projects/Results/Logs 각 폴더 내에 Project 이름으로 지정한 폴더 가 생성됩니다. 한번 지정된 Project 이름은 편집이 불가합니다.
- Server folder : [WorkingDirectory]{Projects/Logs/Results} 경로에 생성 예정인 Project 폴더이며, Input project name의 입력값을 따릅니다. 해당 항목은 사용자가 별도로 작성 할 수 없습니다.
- **Description** : 생성 할 프로젝트에 대한 Description을 작성하는 Text box입니다. Description 항목은 Edit Project를 통해 편집이 가능합니다.

Project 편집/삭제

Project 편집하기

1. Server Explorer -> Projects에 추가된 Project들 중 편집할 Project를 선택하여 마우스 오른 쪽 버튼 클릭하여 **[Edit Project]** 를 선택합니다.



2. 편집 가능한 항목은 Description 항목 뿐입니다. Description 편집이 완료되면 **[OK]** 버튼을 클릭하여 저장합니다.

Project 삭제하기

1. Server Explorer -> Projects에 추가된 Project들 중 삭제할 Project를 선택하여 마우스 오른 쪽 버튼 클릭하여 [Delete Project]를 선택합니다.

2. Confirm 팝업창에서 **[OK]** 버튼을 클릭하면 Project가 삭제되며, [WorkingDirectory]\Projects에서 선택된 Project 폴더 및 파일들이 삭제 됩니다.

Confirm	×
Do you want to delete project 'Test'? 📝 Remove all history of project	
	OK Cancel

• Remove all history of project : Project 삭제 시 과거 이력들도 같이 삭제 할 여부를 체크 합니다. 체크박스에 체크 되어 있을 시 Project가 삭제가 될 때 DB 정보 및 Changeset 폴 더에 있는 관련 파일들이 삭제됩니다.

Mozart Server간 Project 동기화

개요

MOZART Server에 대한 Failover 구성을 위해 Backup 서버를 구축 시 Main(운영) 서버에서 Backup 대상이 되는 Project, Job, Trigger를 Main(운영) 서버와 동일하게 설정을 해야합니다. 즉, Main(운영) 서버를 처음 구축 하기 위해 진행했던 절차들을 Backup 서버에서도 동일하게 진행을 하게 되며, 이로 인하여 발생 가능한 인적 실수, 이중작업에 의한 준비 시간 증가 등, 여러 단점들이 있습니다.

MOZART Server에서는 이러한 단점들을 보완하기 위해 Failover 구성 시 Backup 서버 구축 을 원활하게 하기 위해 1) Main(운영) 서버에 있는 Project/Job/Trigger를 Backup 서버로 복 제하는 기능 및 2) 운영 중에 Main(운영) 서버에 Update가 있을 시 이를 Backup 서버에도 동 기화 할 수 있는 기능 그리고, 3) Main(운영)-Backup 서버 간 동기화에 대한 이력 관리 기능 들을 제공합니다.

기능 동작 방식 및 사용 방법

Main(운영) 서버에 Project/Job/Trigger가 있다고 가정하에 Failover를 위해 Backup 서버 구 축은 다음과 같이 진행 할 수 있습니다.

1. Backup Server 의 MMC에서 Server Explorer에서 동기화 대상이 되는 Main Server를 연결 합니다.

2. Backup Server에서 동기화 시킬 Project 를 생성합니다.(Ex.SimFailover)



- 3. SimFailover Project에서 우측 마우스를 눌러 보조 메뉴에서
 Synchronize Project
 노드

 를 선택합니다.
- 4. [Mapping] 버튼을 클릭하여 동기화 메뉴를 활성화합니다.
 - Source Project: 가져올 서버의 정보를 선택합니다.(MainServer)
 - **Mapping Triggers**: 가져올 서버의 Job/Trigger를 선택합니다.

👬 Synchronize Project	- 🗆 X
Source Project :	Mapping
Mapping triggers :	
Cor 🔡 ServerProjectMapDialog —	
Select Server :	
Options MainServer	~
Project :	
● Overt Sched □····································	
O Do no Sim	
O Synch	
Connect	Cancel
Current Changeset: ### Date synchronized : ####	vnchronize Cancel
Pending Changeset: ### Date scheduled: #### Status: ###	

5. 동기화 대상 Project를 선택하였다면 동기화를 진행 할 Job/Trigger를 선택 및 Option을 설 정합니다.

- **Overwrite all**: 선택한 Job/Trigger의 Schedule 및 Argument 모드를 Mapping된 서버와 동기화를 합니다.
- **Do not overwrite**: 동기화 하지 않습니다.
- Synchronize all, except for checked items: 선택된 Job/Trigger의 Schedule Time을 제 외한 나머지를 다 동기화합니다.

🧱 Synchronize Pro	ject —	□ ×
Source Project : Mapping triggers :	MainServer@Sim Sim@Trigger_Sim	Mapping
Comment:		
Options Sync So	thedule	
Overwrite all		
Do not overwrit Synchronize all	te . except for checked items	
U Synchronize dii,		
Current Changeset Pending Changese	: ### Date synchronized : #### Synchronize t: #### Date scheduled: #### Status: ####	Cancel

6. 동기화를 할 Job/Trigger 설정이 완료 되면, 동기화를 진행 할 Schedule, 즉 Source Server 로부터 Backup Server로 Commit/Deploy 할 시간을 설정합니다.

🔢 Synchronize Project				—	
Source Project : MainServ Mapping triggers : Sim@Trig Comment:	er@Sim ger_Sim				Mapping
 Update Now Update after specific time No schedule yet 	Related Type: Related Items: Kill executing	Job (NONE) triggers	✓		
Current Changeset: #### Pending Changeset: ####	Date synchronized Date scheduled:	: #### #### Status:	####	nchronize	Cancel

7. 모든 설정이 완료되면 **[Synchronize]** 버튼을 클릭하여 동기화를 진행합니다.

8. Main(운영) - Backup Server간 동기화를 진행하게 될 시 언제 어떠한 Project/Job/Trigger 가 어떻게 동기화가 되었는지 확인 할 필요가 있습니다. MOZART Server에서는 Server간 동 기화 이력을 대상 Backup 서버의 SyncProject 통해서 조회가 가능합니다.

🧭 Mozart Management Console 2							
File Window Help							
Server Explorer 🛛 📮 🗙	Lo	calServer/Sync	:Projects 🗙				
20003666	Re	fresh Cance	el				
 Image: Server Explorer Image: LocalServer 	Dr	rag a column header here to group by that column					
🗸 📑 Projects		Changeset	Project	Source Project	Job Triggers	Sync Time	Comment
🗸 📰 failover	٩						
📴 Files	Þ	3	failover	MainServer@Sim	Sim@Trigger_Sim	2019/03/13 19:05	[sync]
E Logs	*						
 Hesuits Jobs Triggers SC_ChangeSet System Log Backup Monitor Performance Shortcuts SyncProjects Users MainServer 							

- **Changeset**: 동기화 식별 또는 Changeset ID입니다. ID가 같고 Project가 다를 시 이는 Multi로 동기화가 되었다는 의미입니다.
- **Project**: 동기화가 진행이 된 대상 Project입니다.
- **Source Project**: 동기화 Source Project입니다.
- Job Triggers: 동기화한 Job과 Trigger 이름입니다.(구분자 @)
- **Sync Time**: 동기화 완료 상태입니다.
- **Comment**: 동기화에 대한 내용입니다. Status가 Fail 시 Error Message가 기록이 됩니다.

운영 중 엔진 로직, Schema, DataAction 등의 변경 사항이 생기는 경우 해당 dll 파일 및 vModel 파일이 업데이트 되어야 합니다. 이때 Backup Server가 존재하는 경우 Main(운영) Server 외 Backup Server에도 대상 파일들을 배포해야합니다. 위와 동일한 절차를 통해 Main(운영) Server의 Update된 Project/Job/Trigger를 Backup Server로 Pull 할 수 있습니다.

Job Scheduler 사용법

다음은 MOZART Management Console(이하 MMC) 을 사용하여 Job Type 및 Job 을 관리하고 작업에 대한 실행을 예약하고 관리하는 방법을 설명합니다.

- Server 의 등록/관리 : MMC 를 통해 관리할 MOZART Server 등록하고 관리하는 방법을 설명합니다.
- Project의 등록/관리 : 대상 Server에서 등록할 수 있는 Project를 등록하고 관리하는 방법 을 설명합니다.
- Job Type 등록/관리 : 대상 Server 에서 실행할 수 있는 Job Type 을 등록하고 관리하는 방법을 설명합니다.
- Job 등록/관리 : : 대상 Server 에 Job 을 등록하고 해당 Job 의 실행 현황을 모니터링하 는 방식을 설명합니다.
- Trigger 등록/관리 : Trigger 를 등록하고 실행 현황을 모니터링하는 방식을 설명합니다.
- Shortcut 등록/관리 : Shortcut 을 등록하고 Server 를 탐색하는 방식을 설명합니다.

Job Type 관리방법

Job Type 은 Job Scheduler 에 의해 실행될 수 있는 작업 유형입니다. 작업 유형에는 전자 메 일 메시지 보내기 및 프로그램 실행과 같은 일반유형과 MOZART 에 의해 개발된 Simple Task, Model Task, Collaboration Task 와 같은 MOZART 전용의 작업 유형이 있습니다.

Model Task 나 Collaboration Task 의 경우에는 실행대상 Model 과 인수의 설정에 따라서 실 행 내용이 변경되므로 Job Type 이 사전에 정의되어 있으며, 일반형태의 작업유형 또한 사전 에 Job Type 이 정의되어 있습니다. 신규 정의가 필요한 Job Type 은 Model 에 기반을 두지 않는 Simple Task 에 국한됩니다.

Job Type 은 Job 조회 화면의 **Edit Job Type** 메뉴를 통해 추가, 수정, 삭제와 같은 관리업무를 수행할 수 있습니다.

Job Type 조회하기

1. Server Explorer 에서 조회하고자 하는 대상 Server 를 선택합니다.

2. Jobs 노드에서 오른쪽 마우스 버튼을 이용하여 [Manage Job Types] 메뉴를 클릭합니다.

3. Job Type Manager dialog 창에서 기 등록된 Job Type 과 시스템에서 사전 정의된 Job Type 을 조회할 수 있습니다.

💀 Job Type Manager			
Name	Description	Category	
Ssendmail Smodel Sexec Scola	sends an e-mail with the configured content to the c Mozart model based task executing native executables in a separate process. Mozart linked model base collaboration task	System System System	
Add	Edit Delete		
			Close

▲ 등록된 Job Type 중 Job Type Name 에 \$ 표시가 된 Job Type 은 시스템에서 제공하는 Job Type 으로 사용자가 수정 및 삭제를 할 수 없습니다.

4. 시스템 Job Type 이 아닌 경우 더블 클릭하거나 하단의 **[Edit]** 버튼으로 등록된 Job Type 내용을 조회하거나 수정할 수 있습니다.

Job Type 등록하기

1. 신규 Job Type 을 등록하기 위해서는 Server Explorer 에서 대상 Server 를 선택합니다.

2. 대상 Server 의 Jobs 노드에서 오른쪽 마우스 버튼을 이용하여 [Manage Job Types] 메뉴 를 클릭합니다.

3. 하단의 [Add] 버튼을 클릭합니다.

4. Job Type 정의 다이얼로그 창이 열립니다.

5. Job Type 정의 다이얼로그의 **Definition Tab** 에 내용을 입력합니다.

rget Definition	ents	X
Category Title Guid	727C1D44-2559-439E-B39B-49E28179AB3E	
Assembly Type Configuration File Private Path		··· ···
Description		
Location		
File name	Target 0	
	OK Cancel	

- Title : Job Type 명을 정의합니다. UI 에서 Job Type 을 정의하는 구분자로 표시됩니다.
- Guid : Job Type 을 유일하게 정의할 수 있는 GUID 를 입력합니다. Text Box 우측의 [...] 버튼을 누르면 신규 GUID로 자동 업데이트됩니다.
- Assembly : MOZART Framework 에서 실행이 가능한 ITask 를 구현한 Assembly/Type 을 지정합니다.
 - Type Text Box 우측의 [...] 버튼을 누르면 Select a Type Dialog 가 활성화됩니다.
 - 신규로 제작한 DII 의 Type 을 선택하려면 **[Load an Assembly]** 버튼을 클릭하여 DII 파일을 선택합니다.
 - 선택된 Assembly 에 포함된 ITask 를 구현한 항목이 Tree 에 표시됩니다.
 - 여기서 등록할 Job Type을 선택하고 **[OK]** 버튼을 클릭합니다.
 - Definition Tab 의 Assembly, Type Text Box 가 선택한 Type 정보로 채워지면 정상적 으로 선택된 것입니다.

- **Type** : 상기 Assembly 선택할 때 함께 선택됩니다.
- Configuration File (Optional): Job Type 의 실행 Log 기록방식 등을 정의한 Config 파일 을 정의할 수 있습니다. Config 파일을 등록하기 위해서는 사전에 해당 Config 파일을 Server 의 Working Folder 에 복사한 뒤, 우측의 [...] 버튼을 통해 Server 상의 해당 Config 파일을 지정합니다.
- Private Path (Optional): Private Path는 해당 Job Type 의 Assembly 파일 및 관련 파일 이 위치할 폴더를 의미합니다. 지정이 되지 않은 경우에는 Working Folder 를 사용합니 다. Working Folder는 Server 설치시점에 지정하며, MMC를 통해서는 Server 의 Working Folder 만 접근이 가능합니다.
- **Description (Optional)** : Job Type 에 대한 설명을 입력합니다.

6. Definition 입력이 끝나면 **Arguments Tab** 을 선택하고 Assembly 에서 지정한 Task 의 입 력 Arguments 를 정의합니다.

Та	get Definit	ion		The second se		and a process	System .	
	Definition	Argu	iments					
	Category		Name	Caption	Туре	Intial Value	Value Range	Description
*								
144	44 4 Rec	ord 0	of0 ▷ ▷ ▷ + -	- √ x 4				•
							(OK Cancel

Get Started

7. 입력된 Arguments 는 Job 및 Trigger 등록시점에 해당 Arguments 의 값을 입력하여 Job Type 이 동작하는 방식을 설정할 수 있습니다.

Job Type 삭제하기

1. 신규 Job Type 을 등록하기 위해서는 Server Explorer 에서 대상 Server 를 선택합니다.

2. 대상 Server 의 Job 노드에서 오른쪽 마우스 버튼을 이용하여 [Manage Job Types] 메뉴를 클릭합니다.

3. Job Type 리스트에서 삭제 대상인 Job Type 을 선택합니다.

4. 하단의 [Delete] 버튼을 클릭합니다. 시스템이 제공하는 Job Type 은 삭제할 수 없습니다.

Job 관리

Job 은 Trigger 에 의해 실행될 Job Type 과 Job Type 의 Arguments 값을 특정하여 해당 Job Type 이 작업을 수행하는 방식을 정의한 Data 입니다. Server Explorer 의 Server 별 Job View 를 통해 등록된 Job 을 조회할 수 있으며 화면의 상단메뉴를 통해 Job 의 추가, 수정, 삭제와 같은 작업을 수행할 수 있습니다. Job 에서 설정된 Arguments 는 Trigger 편집 시 Job 의 기본 Parameters 설정값으로 사용됩니다. 즉, Trigger 편집 시에 값을 변경하지 않는 경우 Job 에 설정된 Parameters 값으로 Job 이 실행됩니다.

- Job 등록/수정 방법 : 신규 Job 을 등록하거나, 기존 등록된 Job 의 등록 내용을 수정하는 방법을 설명합니다.
- Job 삭제 방법 : 등록된 Job 을 삭제하는 방법을 설명합니다.

Job 등록하기

1. Server Explorer 에서 조회하고자 하는 대상 Server 를 선택합니다.

2. 대상 Server 의 Jobs 노드를 더블 클릭하여 Jobs 조회 화면을 활성화합니다.

Jobs ×		
Add Edit	Remove Refresh	
Drag a colum	n header here to group by t	hat column
JobType	Name	ConcurrentExecution Description
٩		
\$model	SimpleMfg	
\$model	testmodel	\checkmark
-		

3. 상단 메뉴바에서 [Add] 메뉴를 클릭합니다.

4. New Job Dialog 에서 Basic 및 Parameter Tab 에 정보를 입력합니다.

5. Basic Tab 의 내용을 먼저 입력합니다.

💀 New Job			
Basic Pa	rameters		
You must s	specify what action this task will perform.		_
Job name:	TestJob		
Description	n: This is test job		
Job Type:		•	
	Disallow Concurrent Execution		
	No options for setting.		
	ОК	Cancel	ן ר

- Job Name : Job 명을 입력합니다. 입력된 Job 명은 Job 을 유일하게 구분하는 구분자로 사용됩니다.
- **Description** : Job 에 대한 설명을 입력합니다.
- Job Type : Job 이 수행할 Job Type 을 콤보박스에서 선택합니다. Job Type 중 Model Task 를 기반으로 하는 \$model, \$cola 두 가지 Job Type 은 Model 의 선택 및 Model 실행 옵션에 따라 설정해야 하는 Parameters 가 변경되어야 하므로 이를 결정할 수 있는 Job Setting 입력창이 표시됩니다.

👬 New Job			• ×					
Basic Parameters								
You must specify what action this task will perform,								
Job name:	TestJob							
Description:	This is test job							
Job Type:	\$model		-					
	Disallow Concurrent	t Execution						
	Job Setting							
	Project :	Server Test						
	Model file :	Model, vmodel 🔹						
	Model dll file :	ServerTest, dll 🔹						
	Configuration file :	• • • • • • • • • • • • • • • • • • •						
	Log dir :	Server Test						
	Additional run count :	: 0						
		OK Ca	ncel					

- **Project** : Model Task가 실행할 Model 파일 및 Model을 실행 할 DLL 파일 정보를 가지고 있는 Project를 지정합니다. 드롭다운 리스트를 조회하면 선택 가능한 Project의 리스트 가 나타납니다.
- Model file : Model Task 가 실행할 Model 파일을 지정된 Project에서 지정합니다. 하나 의 Project 에 여러개의 Model 파일이 있는 경우 리스트에 있는 Model 파일 하나만 선택 가능합니다.

Job Setting	
Project :	Test 🔹
Model file :	▼
Model dll file :	FabModel, vmodel Model, vmodel

• Model dll file : Model 을 실행할 DII 파일을 지정합니다. 사전에 Server 에 파일을 업로드 해야 합니다.
- Configuration file : Model 수행 시 참조할 Configuration 파일을 지정합니다.
 Configuration 파일에는 해당 Model 의 실행 Log 파일에 대한 로그 Key 및 Log 폴더 등을 추가적으로 지정할 수 있습니다.
- Log dir : Job 실행 Log 를 남길 폴더를 지정합니다. 기본적으로 지정한 Project의 Logs 폴더로 지정이 됩니다. 변경 할 시 우측의 [...] 버튼을 사용하여 폴더를 선택할 수 있으며, 표시는 [Working Directory / Logs] 폴더를 기준으로 표시됩니다.

Browse For Folder	- • •
WorkingDirectory Backup Changesets Jobs Changesets Jobs Colla_Test Fab_Planning Failure_Test HKMA_Planning Hynix_Planning Link MyProject Server Test MyProject MyProject System Models	
Folder: WorkingDirectory\Logs\Test	
New Folder <u>O</u> K	<u>C</u> ancel

- Additional run count : 동일 Model 을 1회 이상 실행하고자 할 경우 설정합니다. 기본값 은 0입니다. 설정한 경우 Parameters 에 해당 차수별로 사용할 Arguments 설정 파일을 지정할 수 있는 Arguments 가 생성됩니다.
- **Collaboration Count** : \$cola 형식의 Job Type 을 선택한 경우에만 입력할 수 있도록 표시 됩니다. Main Model 구동 시 Collaboration 대상이 되는 Job Type 을 설정할 수 있도록 Parameters 에 설정한 차수별로 Job Type 을 설정할 수 있는 Arguments 가 생성됩니다.
- **Disallow Concurrent Execution** : MOZART Server 가 Job 을 동시에 여러 개 실행시킬 수 없도록 설정할 수 있습니다.

6. Basic Tab 에서 Job Type 에 대한 선택이 완료되면 Parameters Tab 에 해당 Job Type 에 서 설정할 수 있는 Arguments 리스트가 표시됩니다. 이때 실행할 Job 의 특성에 따라 Arguments 의 값을 입력합니다.

+ 전자 메일보내기, 프로그램 실행 및 사용자에 의해 등록된 Job Type 이 아닌 Model Task, Collaboration Task 를 선택한 경우에는 Basic Tab 에서 입력한 Model 의 Arguments 가 함께 표시됩니다. 아래의 상단 구분에 나와있는 Parameters 는 Model 에서 별도로 만든 Input Arguments 이며, 하단의 extentedProperties 는 시스템에서 자체적으로 Model 의 실행 및 후 속 처리에 대한 Option 을 제공하기 자동으로 추가된 Arguments 입니다. 해당 Arguments 에 대한 설명은 Extended Arguments 를 참고하시기 바랍니다.

New Job	
Basic Parameters	
🔺 (General)	
throwException	
#use-database	
#save-database	
∡ z - extendedProperties	
#daction-excludes	
#daction-includes	
#daction-excludes/in	
#daction-includes/in	
#start-time.AdjustMinutes	0
#overwrite-result	
#model-file	Projects\Test\Model.vmodel
#model-dll	Projects\Test\ServerTest.dll
#model-config	
#experiment	Experiment 1
#db-to-file	
#file-to-db	
#db-includes	
#db-excludes	
#zip	\checkmark
#zip.Path	Test
#zip.FileNamePostfix	
#zip.FileNameTemplate	
#zip.UpdateToRecent	
#more-runs	0
#run-index	
#log-level	
#log-dir	Test
#dataSource-set-default	K1=V1;K2=V2
#performance-profiling	
₩ *	/ x 4
	OK Cancel

Job 수정하기

1. Server Explorer 에서 조회하고자 하는 대상 Server 를 선택합니다.

2. 대상 Server 의 Jobs 노드를 더블 클릭하여 Job 조회 화면을 활성화합니다.

3. Job 리스트에서 수정하고자 하는 Job 을 선택합니다.

4. 상단 메뉴바에서 [Edit] 메뉴를 클릭합니다.

5. *"Job 등록하기"* 에서 편집한 방식과 동일하게 항목을 수정합니다.

+ Job 에 대한 Trigger 가 생성된 이후에 Job 에서 Parameters 값을 수정하여도 해당 Job 형식으로 생성된 Trigger 의 설정값이 자동으로 변경되지 않습니다. 이점 유의하시기 바랍니 다.

Job 삭제 방법

1. Server Explorer 에서 조회하고자 하는 대상 Server 를 선택합니다.

- 2. 대상 Server 의 Jobs 노드를 더블 클릭하여 Jobs 조회 화면을 활성화합니다.
- 3. Job 리스트에서 삭제하고자 하는 Job 을 선택합니다.

4. 상단 메뉴바에서 [Remove] 메뉴를 클릭합니다.

▲ Job 을 삭제하기 위해서는 해당 Job 의 Trigger 를 먼저 삭제해야 합니다. 삭제 대상 Job 의 Trigger 가 있는 경우에는 Trigger 정보가 있어서 삭제할 수 없다는 메시지가 표시됩니다.

Config 파일 설정방법

Config 파일로 Model 실행 Log 파일을 저장할 폴더를 지정할 수 있습니다. 다음은 Config 파 일 작성 방식을 설명합니다.

먼저 해당 Model 에 대한 Config 파일을 생성합니다. 일반 Text 파일을 생성하고 확장자는 ".config" 로 저장합니다. 파일내용은 아래와 같은 방식으로 작성합니다. xml 파일로 작성하 며, 세션에 Key 로 추가 합니다.

작성 가능 Keyword

#log-dir : Job 수행 Log 를 기록할 폴더를 지정하기 위한 Key 입니다.

+ 값 설정방식 : Log 파일이 생성될 폴더를 절대경로로 작성하거나, Server 의 Working Directory/Logs 폴더 기준의 상대경로로 작성합니다. 작성 예시 :



상기예시와 같이 작성한 경우 Working Directory/Logs/AmodelLog 폴더에 Log 파일이 생성 됩니다.

작성된 파일을 Server 의 특정 폴더에 업로드합니다. 일반적으로 Model 파일과 동일 폴더에 저장합니다. Job 설정 시 Config 파일을 설정합니다.

Job Setting		
Model file:	$D:\MozartServer\mozartModelsSimpleMf$	
Model dll file:	D:\MozartServer\vms\mozart\Jobs\SimpleMfg.d	
Configuration file:	D:MozartServervmsmozartModelsSimpleMf	
Additional run count:	1	

상기와 같이 설정한 경우 해당 Job 의 Task, Persist Log 가 지정 폴더에 생성됩니다. TSK Log 파일의 Log Key 는 "model-name" Argument 의 입력값이 되며, 설정되지 않은 경우 기본 Key 는 "TSK" 입니다.

More Run 설정 방법

1. More Run 설정은 한 번의 Job 실행으로 Model 을 여러 번 수행하는 것이 필요한 경우 사용 합니다. 예를 들어 특정 시간에 Data 를 Download 받고 Model 을 한번 실행한 후 Model 실행 결과를 사용하여 Model 을 한 번 더 실행해야 하는 경우 사용할 수 있습니다.

이러한 More run 의 설정은 Job 설정 시 할 수 있습니다. 아래는 More Run 설정 방법입니다.

2. Job 의 Basic Tab 에서 Job Type 을 \$model 혹은 \$cola 로 선택합니다. (두 가지 Job Type 에서만 More Run을 지원합니다.) Job Type 선택 시 Job Setting 에 표시되는 Additional run count 를 설정합니다. 예를 들어 Additional run count 값을 '1'이라고 설정하면 1번 더 실행하여 총 실행횟수가 2회가 되는 구조입니다.

Job Setting		
Model file:	Models\2015.03.24_mhmonitor\Model.vmodel	
Model dll file:	Jobs\2015.03.24_hmmonitor\HMKAInputMonito	
Configuration file:		
Log dir:	Monitor	
Additional run count:	1	

3. 추가 실행 때 사용할 Arguments 를 Setting 하기 위해서는 Parameters Tab 으로 이동합니다. extendedProperties 의 가장 마지막 부분에 추가 실행에 대한 Parameters 설정을 입력할 수 있는 Extended Argument 가 추가되어 있습니다. 아래 그림에서와 같이 #more-runs = 1 로 설정되며, #more-config-1 이 생성됩니다.

🖳 New	Job	
Basic	Parameters	
#use	e-database	
#sav	ve-database	
#mo	odel-file	Models\model.vmodel
#mo	odel-dll	
#mo	odel-config	
#ext	periment	Experiment 1
#db	-to-file	
#db	-includes	
#db	-excludes	
#zip		• • • • • • • • • • • • • • • • • • •
#zip	.Path	
#zip	.FileNamePostfix	
#zip	.FileNameTemplate	
#zip	.UpdateToRecent	
#mo	ore-runs	1
#run	n-index	
#log	ı-dir	
#dat	taSource-set-default	K1=V1;K2=V2
#mo	ore-config-1	··· 🛞
		· · · · · · · · · · · · · · · · · · ·
144 44	Record 27 of 27 + ++ ++ + +	- A 🗸 X 4 🛛 🕨
		OK Cancel

4. 위의 그림에서 #more-config-1 의 [...] 버튼을 클릭하면 Model 에서 입력한 Input Arguments 에 해당하는 Argument 의 값을 입력할 수 있는 창이 열립니다. more-run 수행 시 사용할 입력 Argument 값을 입력합니다. extended argument 중 추가 실행 시 변경이 가능한 Arguments 역시 변경할 수 있습니다. 예를 들어 추가 실행 시에 #save-database 옵션을 꺼서 Output Data 를 DB 에 쓰지 않게 한다던지 하는 등의 처리를 수행할 수 있습니다.

🖳 Parameter S	let	
Name:	<none></none>	
Model file:	Models\2015.03.24_mhmonitor\Model.vmodel	
🔺 z - exte	ndedProperties	
#daction-ex	cludes	
#daction-inc	ludes	
#daction-ex	cludes/in	
#daction-inc	ludes/in	
#use-databa	ase	
#save-datab	ase	
#db-include	5	
#db-exclude	95	
#log-dir		
He ee e Re	cord 1 of 9	4
	ОК	Cancel

Server DLL 버전과 상이한 모델 수행하는 방법

Server를 운영하게 될 시 현장에 따라 안정적으로 수행이되는 Task의 DLL 버전들이 각기 다 릅니다. 안정적으로 운행이 되고는 있지만 S/W의 특성상 신규 기능 추가 등에 사유에 따라 제품 업그레이드가 불가피한 경우들이 많습니다. 그러나 제품 업그레이드를 하게 될 시 현재 운영중인 Task DLL 및 Model과의 호환성이 안 맞을 수도 있습니다. 제품 업그레이드 이전 까 지는 호환성의 문제를 확인하기 어려운 부분들이 많으며 호환성을 확인하기 위해 제품 업그 레이드에 소요되는 시간, 또한 문제가 발생하여 다시 호환이 잘 되던 버전으로 Rollback에 소 요되는 시간 등, 즉각 대응의 문제와 비효율적인 시간 소모들이 많습니다.

MOZART Server에서는 위에 명시된 발생 가능한 문제들을 예방 및 완화를 하고자 Server 머 신에 설치된 MOZART Library DLL과 Task DLL의 버전이 상이하여도 Task가 잘 수행이 되도 록 구성이 되어 있습니다. 상이한 버전의 Task를 수행하기 위해서는 MMC를 통해 사용자가 설정을 해야 합니다. 아래 그림은 MOZART Server에서 설치된 Server DLL과 상이한 버전의 Task가 수행되는 방법에 대한 기본 개념도입니다.



상기의 그림과 같이 구성을 하려면 Server 머신에는 **2018.2.113.0** 버전 이상의 MOZART Server가 설치 되어 있어야 합니다. 일반적으로는 Task DLL 및 Model를 Server에서 수행하기 위해서는 Server 설치된 DLL과 동일한 버전의 Client에서 Task DLL 및 Model 파일이 생성이 되어야 합니다. 그러나 Client의 변경 없이 상이한 버전의 Server에서 Task DLL 및 Model를 수행하기 위해서는 수행하고자 하는 Task DLL의 호환되는 버전의 Server DLL를 WorkingDirectory에 생성을하여 MMC에서 해당 Task 및 Model이 WorkingDirectory에 위치 한 Server DLL를 통해 수행 할 수 있도록 설정을 하면 됩니다. 다음은 MMC를 통해 설정하는 방법에 대해서 기술합니다.

☑ MMC를 통해 설정하기 이전 Task DLL 및 Model의 Host 역할을 할 Server DLL 파일들이 WorkingDirectory에 옮기는 사전 작업이 필요합니다. 예를 들어 현재 Server 머신에 설치된 버전이 2018.2.113.0 버전이고 Task DLL 및 Model이 수행되던 버전이 2017.1.108.0이었을 경우, 2018.2.113.0 버전 MOZART Server 설치 전 2017.1.108.0 Server 폴더 (예: C:\Program Files\VMS\Mozat\Server)를 WorkingDirectory에 복사를 합니다.

MOZART Management Console를 통해 설정하는 방법

1. Server Explorer에서 대상 서버를 선택합니다.

2. 대상 서버의 Jobs 또는 Triggers 노드 더블 클릭하여 Jobs/Triggers 창을 활성화 합니다.

3. Task의 Host가 변경이 필요한 Job 또는 Trigger를 선택합니다.

4. Job을 선택하였으면 [Parameter] 탭으로, Trigger를 선택하였으면 [Target Job] 탭으로 이 동을 합니다.

🔡 Edit Job		_		×
Basic Parameters				
				_
#db-to-file				-
#file-to-db				
#db-includes				
#db-excludes				
#zip	\checkmark			
#zip.Path	111_FAB			
#zip.FileNamePostfix				
#zip.FileNameTemplate				
#zip.UpdateToRecent				
#more-runs	0			
#run-index				
#log-level				
#log-dir	111_FAB			
#dataSource-set-default	K1=V1;K2=V2			
#performance-profiling	✓			
#host-dir	111_Server		•	
141 44 4 Record 47 of 47 b bb bb + -	A J X 4			
Record 47 of 47 in a mini				
		OK	Canc	el

5. Argument 리스트에서 **"#host-dir**"를 찾아 **[...]** 버튼을 클릭하여 Task 수행 할 Host의 경로 를 지정합니다.

6. **[OK]** 버튼을 클릭합니다.

Trigger 관리

Trigger 는 실행대상 Job 과 Job 의 실행 조건 정의할 수 있는 Job Scheduler 의 관리 요소입 니다. 하나의 Job 을 실행하는 방식에 따라 여러 개의 Trigger 를 등록할 수 있습니다. 예를 들 어 동일한 Job 을 매주 월요일에는 오전 7시에 수행하고, 금요일에는 오후 9시에 수행하고자 한다면 해당 Job에 대한 Trigger 를 두 개 설정하면 됩니다. Trigger 실행 조건으로는 일정 시 간, 주기에 종속적인 시간 기반의 조건과 특정 Trigger 가 실행되는 이벤트에 종속적인 이벤트 기반 조건 두 가지가 있습니다.

- Trigger 등록/수정 방법 : 신규 Trigger 를 등록하거나, 기존 등록된 Trigger 의 등록 내용 을 수정하는 방법을 설명합니다.
- **Trigger 복사 방법**: 등록된 Trigger를 다른 Trigger Name으로 설정하여 복사하는 방법을 설명합니다.
- Trigger 삭제 방법 : 등록된 Trigger 를 삭제하는 방법을 설명합니다.

Trigger UI 설명

MOZART Management Console의 Triggers의 UI에 대한 설명입니다. Triggers 탭에서는 Trigger 등록/편집/삭제/복사 등의 작업을 수행할 수 있으며 그 외에 등록된 Trigger의 이력, 수행 로그 등도 조회가 가능합니다. 아래는 Triggers 노드를 활성화 하였을 시 보이는 UI입니 다.

상단메뉴 바 및 Trigger 리스트

Trigger 등록/편집/삭제/이력 조회 등을 할 수 있는 UI입니다.

- Add : Trigger를 등록하기 위해 사용하는 메뉴입니다.
- Edit: Trigger 리스트에 등록된 Trigger를 편집하기 위한 메뉴입니다.
- **Remove** : Trigger 리스트에 등록된 Trigger를 삭제하기 위한 메뉴입니다.
- Copy : Trigger 리스트에 등록된 Trigger를 복사하여 리스트에 추가하기 위한 메뉴입니다.
- **Refresh** : Trigger의 Next Time/State/StartTime/EndTime 및 Execution History등의 정 보를 새로고침 하기 위한 메뉴입니다.
- View History : Trigger 리스트에 등록된 Trigger의 과거 이력을 조회하기 위한 메뉴입니다. View History를 통해 어떤 사용자가 어떤 작업을 진행을 하였는지 확인이 가능합니

다.

• **Open Log Folder** : Trigger 리스트에 등록된 Trigger의 로그 파일들을 조회하기 위한 메뉴 입니다.

Log Files 창

\WorkingDirectory\Logs[Project Name]의 Trigger 수행 로그 파일들을 조회 할 수 있는 영역 입니다. Execution History의 Search Option의 검색 범위 내에 포함이 되면 파일 리스트들이 표시 됩니다. Triggers의 Log Files 영역에서는 최근 실행된 10개의 파일만 표시되며, 전체 리 스트를 보기 위해서는 [**Open Folder**]를 클릭하여 Logs 탭을 활성화 할 수 있습니다.

Log Files				
Open Folder				
Name	Date	Size	Attributes	
task-Fab6-20170519-113345.log	2017-05-19 11:36:05	4 kb	-2	
task-Fab6-20170519-112520.log	2017-05-19 11:27:40	4 kb	-a	
task-Fab6-20170504-134719.log	2017-05-04 13:55:37	4 kb	-a	
task-Fab6-20170504-131719.log	2017-05-04 13:25:50	4 kb	-a	
task-Fab6-20170504-124719.log	2017-05-04 12:55:46	4 kb	-a	
task-Fab6-20170504-121719.log	2017-05-04 12:26:14	4 kb	-a	
task-Fab6-20170504-114719.log	2017-05-04 11:55:37	4 kb	-a	
task-Fab6-20170504-093633.log	2017-05-04 09:55:42	4 kb	-2	
task-Fab6-20170502-190633.log	2017-05-03 02:41:13	13 kb	-a	
task-Fab6-20170502-183633.log	2017-05-02 22:37:03	7 kb	-a	

Execution History 창

Execution History 창은 검색 조건에 따라 지정된 Trigger의 Monitoring 정보를 표시합니다. 해당 내용은 Monitoring 노드에서도 확인이 가능합니다.

사용 방법은 다음과 같습니다.

Search Option에 숫자를 입력합니다. 검색 조건은 현재로부터 최근 N일의 정보를 표시합니다. (i.e 현재로부터 최근 10일 -> Search Option에 '10' 입력) [Query] 버튼 클릭

E	ecution History							
S	earch Option: 30	Days Query						
Di	rag a column heade	er here to group by that colu	mn					
	TriggerName	Scheduled	Start	End	Elapse	Status	Result	Message
Ŷ								
Þ	Fab	2017-04-11 15:36:31	2017-04-11 15:36:31	2017-04-11 15:39:26	00:02:55	🥥 Complete	SUCCESS	
	Fab	2017-03-29 09:56:33	2017-03-29 09:56:33			Aborted		
	Fab	2017-03-28 15:10:33	2017-03-28 15:10:33	2017-03-28 16:03:51	00:53:18	🥥 Complete	SUCCESS	
	Fab	2017-03-27 19:55:20	2017-03-27 19:55:20	2017-03-27 19:58:41	00:03:21	🥥 Complete	SUCCESS	
	Fab	2017-03-27 19:25:26	2017-03-27 19:25:26	2017-03-27 19:25:37	00:00:11	🥥 Complete	FAIL	${\tt System.AppDomainUnloadedException: \ \ \ \ \ \ \ \ \ \ \ \ \ $
	Fab	2017-03-27 19:19:20	2017-03-27 19:19:20	2017-03-27 19:22:41	00:03:21	🥥 Complete	SUCCESS	
	Fab	2017-03-27 15:43:25	2017-03-27 15:43:25	2017-03-27 15:46:08	00:02:43	🥥 Complete	SUCCESS	
	Fab	2017-03-27 14:01:11	2017-03-27 14:01:11			🥝 Aborted		
	Fab	2017-03-27 13:48:01	2017-03-27 13:48:01			🥥 Aborted		
	Fab	2017-03-27 13:19:22	2017-03-27 13:19:22			🥝 Aborted		
	Esh	2017 02 27 12:17:50	2017 02 27 12:17:50			Abortod		

♀ Triggers의 Execution History는 Paging 방식으로 표시를 하여 최초에는 창 크기에 따라10개 단위로 항목들을 표시합니다. 사용자가 스크롤 다운 시 10개 단위로 증가합니다.

Trigger Execution Log 창

Trigger Execution 창에서는 Execution History의 창에서 선택한 Trigger의 수행 시간을 단계 별로 확인 할 수 있는 창입니다. 자세한 내용은 Monitoring에서 확인이 가능합니다.

Tri	igger Execution Log			
	Action	Start	End	Elapse
÷	PERSIST_IN	2017-03-28 15:10:54	2017-03-28 15:13:28	00:02:34
	ENGINE_RUN	2017-03-28 15:13:30	2017-03-28 15:43:26	00:29:56
	PERSIST_OUT	2017-03-28 15:44:22	2017-03-28 15:44:32	00:00:10

Trigger 등록 방법

- 1. Server Explorer 에서 Trigger 등록 대상 Server 를 선택합니다.
- 2. 대상 Server 의 Triggers 노드를 더블 클릭하여 Trigger 조회화면을 활성화합니다.
- 3. 상단 메뉴바에서 [Add] 메뉴를 클릭합니다.
- 4. New Trigger Dialog 에서 Schedule, Target Job, Failure Action Tab 별 정보를 입력합니다.
- 5. Schedule Tab 정보를 입력합니다.

Schedule Target Job Trigger name: TestTrigger2 Settings One Time Start: 2015-01-19 2 = 9:19:21 Daily Daily Weekly Monthly Dependent Advanced Settings Priority: 0 = 0 00:00:00 Retry Interval: 00:00:00 Retry Count: 0 = 0 OK Cancel 	🚪 New Trigger							23
Trigger name: TestTrigger2 Settings One Time Start: 2015-01-19 Daily Daily Weekly Monthly Dependent Priority: Organization Priority: Organization Retry Interval: 00:00:00 Retry Count: OK Cancel	Schedule Targe	t Job Failure /	Action					
Settings One Time Simple Expire: 2015-01-19 2 = 9:19:21 Daily Weekly Monthly Dependent Dependent Priority: 0 = 0 Retry Interval: 00:00:00 Retry Count: 0 = 0 	Trigger name:	TestTrigger2						
One Time Start: 2015-01-19 2 \$\overline\$ 9.19:21 Simple Expire: 2015-01-19 2 \$\overline\$ 9.19:21 Daily 2 \$\overline\$ 9.19:21 \$\overline\$ Weekly Monthly Dependent 0 Advanced Settings Priority: 0 0 0 Stop task if it runs longer than: 00:00:00 Retry Count: 0 V Enabled Trace OK	Settings							
Simple Expire: 2015-01-19 Q = 9:19:21 Nonthly Monthly Dependent Advanced Settings Priority: 0 Stop task if it runs longer than: 00:00:00 Retry Interval: 00:00:00 Retry Count: 0 Trace OK Cancel	One Time	Start:	2015-01-19	· 	9:21			
○ Daily ○ Weekly ○ Monthly ○ Dependent △ Advanced Settings ○ Priority: ○ ♥ ○ Stop task if it runs longer than: 00:00:00 ○ Enabled ○ Trace	Simple	Expire:	2015-01-19		9:21			
○ Weekly ○ Monthly ○ Dependent △ Advanced Settings ○ Priority: ○ 😓 ○ Stop task if it runs longer than: 00:00:00 ○ Enabled ○ Trace	Daily							
○ Monthly ○ Dependent ○ Advanced Settings ○ Priority: ○ ♥ ○ Stop task if it runs longer than: 00:00:00 ○ Enabled ○ Trace	Weekly							
Advanced Settings Priority: 0 - Stop task if it runs longer than: 00:00:00 Retry Count: 0 - Enabled Trace	Monthly Dependent							
Advanced Settings Priority: Stop task if it runs longer than: OC:00:00 Retry Count: OC OK Cancel	O bependent							
Advanced Settings Priority: 0 Stop task if it runs longer than: 00:00:00 Enabled Trace OK								
 Priority: Stop task if it runs longer than: 00:00:00 Retry Count: 0 → Trace 	Advanced Setting	gs						
Stop task if it runs longer than: 00:00:00 Retry Count: 0 Image: Trace OK Cancel	Priority:			0	Retry Interval:		00:00:00	
✓ Enabled ☐ Trace OK Cancel	Stop task if it	runs longer than:		00:00:00	Retry Count:		0	
Trace	✓ Enabled	-						
OK Cancel	Trace							
OK Cancel								
OK Cancel								
						ОК	Cancel	

- Trigger name : Trigger 명칭
- Settings : Trigger 실행 조건 설정
- One Time : 시작 시각에 1회 실행하도록 설정합니다.
- Expire 설정 : Expire 체크상자에 체크하는 경우 설정된 Expire 시각까지만 Trigger 가 발생하게 됩니다.
- Simple : 반복횟수와 반복주기를 설정할 수 있습니다. 반복 주기는 초 단위부터 일 단위 까지 지정할 수 있습니다. Repeat Count 가 지정된 경우 Recur every 에 설정된 주기에 따 라 반복횟수만큼 실행됩니다.

Settings				
🔘 One Time	Start:	2015-03-26		3:35
Simple	Expire:	2015-03-26		3:35
Daily				
Weekly				
Monthly	Recur ever	y:	0.00:00:00	intervals
Dependent	Repeat cou	int:	0	(※ -1 : 무제한)
	🗌 for a du	uration of:	00:00:00	

- Daily : 일 단위 주기로 반복하도록 Schedule 정의가 가능합니다. Simple Type 으로 동일 하게 Schedule 을 수립할 수 있습니다.
- Weekly : 주 단위 주기를 설정할 수 있으며, 주에 실행될 요일을 지정할 수 있습니다.
- Monthly : 월+일 지정, 월 지정+월의 특정순서의 요일을 지정할 수 있습니다.
- Dependent : 이벤트 기반 조건 지정 방식으로 참조 대상이 되는 trigger 를 선택하고 해당 trigger 의 시작/종료 시점 이벤트를 선택할 수 있습니다. Dependent 로 설정된 trigger 는 Target trigger의 실행에 종속적으로 실행이 됩니다.

Settings					
🔘 One Time	Start:	2015-03-26		•	
Simple	Expire:	2015-03-26			
Daily					
Weekly	D-fIT-i				
Monthly	Kererred Tri	gger:		•	
Oependent	Execution ty	/pe:	AtEnd	-	
	Execution d	elay:		00:00:03	

Referred Trigger:	
Execution type:	monitor dependent
Execution delay:	

AtEnd	-
AtEnd	
AtStart	
ReturnIfTrue	

- Referred trigger : 설정하는 trigger의 실행을 결정하는 trigger입니다. 복수의 trigger를 선 택할 수 있습니다.
- Execution type : 참조 trigger 와 연계하는 방식을 정의합니다.
 - AtEnd : 참조 trigger 가 종료되면 시작합니다.
 - AtStart : 참조 trigger 가 시작될 때 같이 시작합니다.
 - ReturnIfTrue: 참조 trigger 의 결과값이 true 를 반환하면 실행합니다. 예를 들어 특정 시스템에서 Data 를 변경한 후 개발한 Job 을 실행해야 하는 경우 혹은 참조 trigger 에 DB 의 특정 Data 를 조회해서 값이 조건을 만족하는 경우 true 를 반환하는 Job 을 설정할 수 있습니다. 이 경우에는 참조 trigger 에서 수행하는 Job 의 Model Task Return 값을 true/false 로 설정해주어야 합니다.
 - ReturnIfTrue 타입 사용예
 - Execution delay : Execution type 에서 설정한 조건이 만족한 상태에서 trigger 시작 까지의 지연시간을 설정합니다.
- Advanced Settings
 - Priority : 여러 개의 trigger 가 동시에 처리를 하려할 때의 실행 우선 순위
 - Stop task if it runs longer than : trigger 에서 실행한 작업의 실행시간이 정해진 시간
 을 초과하는 경우 실행 중지하도록 처리하는 옵션
 - Enable : 해당 trigger 의 사용 여부
 - Retry Interval : trigger 실행 실패 시 재실행 주기
 - Retry Count : trigger 실행 실패 시 재실행 횟수
- 6. Target Job Tab 정보를 입력합니다.

🔡 New Trigger	
Schedule Target Job Failure Action	
Target job: Server Test, Server Test	
You must specify what arguments this task will use, 	Reset to defaults
∡ (General)	<u> </u>
throwException	
∡ z - extendedProperties	
#daction-excludes	
#daction-includes	
#daction-excludes/in	
#daction-includes/in	
#start-time.AdjustMinutes	0
#overwrite-result	
#use-database	\checkmark
#save-database	\checkmark
#model-file	Projects#Server Test#Model.vmodel
#model-dll	Projects#Server Test#ServerTest.dll
#model-config	
#experiment	Experiment 1
#db-to-file	
#file-to-db	
H4 44 4 Record 1 of 28 > >> >> + + - ▲ √ × 4	4
Schedule Now	OK Cancel

- Target Job : trigger 가 실행할 대상 작업(Job)으로 Jobs 에 등록된 Project.Job만 선택될 수 있습니다.
- Arguments : Target Job 에 따라 실행 시 필요한 입력 Parameters 를 설정합니다. 기본적 으로 Job 에 설정된 Parameters 값이 채워집니다. 실행 시에는 trigger 에 설정한 Arguments 값이 사용됩니다.
- 7. Failure Action Tab 정보를 입력합니다. Target Job 입력형식과 같습니다.
 - Failure Action : Target Job 의 실행실패 시에 수행될 Job 으로 신규 기능이 추가되거나, 보완될 때 백업 Job 을 선택할 수 있습니다.
 - Arguments : Failure Action 으로 선택된 Job 의 입력 Parameters 를 설정합니다. Target Job 과 같이 Job 정보에 설정된 Parameters 를 기본값으로 합니다.

Trigger 수정 방법

- 1. Server Explorer 에서 수정하고자 하는 대상 Server 를 선택합니다.
- 2. 대상 Server 의 Triggers 노드를 더블 클릭하여 Trigger 조회 화면을 활성화합니다.
- 3. Trigger 리스트에서 수정하고자 하는 Trigger 를 선택합니다.
- 4. 상단 메뉴바에서 [Edit] 메뉴를 클릭합니다.
- 5. "Trigger 등록하기"에서 편집한 방식과 동일하게 항목을 수정합니다.

Trigger 복사 방법

동일한 Trigger를 Argument 일부만 변경하여 두 Trigger의 결과를 비교하거나 기존에 등록된 Trigger의 명칭 변경이 필요 시 Trigger 복사 기능을 통해 수행이 가능합니다.

- 1. Server Explorer 에서 조회하고자 하는 대상 Server 를 선택합니다.
- 2. 대상 Server 의 Triggers 노드를 더블 클릭하여 Triggers 조회 화면을 활성화합니다.
- 3. Trigger 리스트에서 복사하고자 하는 Trigger를 선택합니다.
- 4. 상단 메뉴바에서 [Copy]를 클릭합니다.
- 5. 복사 될 Trigger의 명칭을 입력합니다.

▲ 복사 될 Trigger의 이름을 이미 존재하고 있는 Trigger의 이름으로 입력하게 될 시 오류가 발생하니 반드시 Trigger 리스트에 없는 이름으로 입력이 되어야 합니다.

Trigger 삭제 방법

1. Server Explorer 에서 조회하고자 하는 대상 Server 를 선택합니다.

2. 대상 Server 의 Triggers 노드를 더블 클릭하여 Triggers 조회 화면을 활성화합니다.

3. Trigger 리스트에서 삭제하고자 하는 Trigger 를 선택합니다.

4. 상단 메뉴바에서 [Remove] 메뉴를 클릭합니다.

Trigger가 참조하는 모델 및 데이를 임시폴더 에 복사 후 실행 기능

서버에서 Trigger가 수행될 때 Trigger에 설정된 vModel 파일이 위치한 폴더(이하 모델 폴더) 가 아닌 임시로 생성한 실행 폴더에서 Trigger를 수행하는 방법이 있습니다. 임시 실행 폴더 를 이용 시 Trigger 구동 방식은 아래와 같습니다.



실행 폴더 사용을 위한 Argument

실행 폴더 사용 여부를 판단하기 위한 #use-run-dir 와 실행 폴더 생성 수를 제어 하기 위한 #max-run-dir Argument 가 추가되었습니다.

Argument DataType Description

#use-run- dir	boolean	실행 폴더를 사용하여 수행할지 여부(default : false)
#max-run- dir	int	실행 폴더 생성 갯수(default : 2)
#use- parent-path	boolean	Dependent Trigger 사용 시 Reference Trigger의 임시실행 폴더 를 사용할지 여부 (default: false)

#use-run-dir = true인 경우, 실행 폴더가 Trigger 수행 시 생성되게 됩니다. 생성되는 실행 폴 더의 위치는 다음과 같습니다.

실행 폴더 위치 : WorkingDirectory\Execution\[Trigger 명]\Temp\[날짜-시간-임의명칭]

max-run-dir 의 값에 따라 서버에서 유지할 실행 폴더 갯수가 정해집니다. (ex. 값이 2 이면, 가 장 최근에 실행 된 2개의 폴더만 유지됩니다.) 만약 사용자가 실행 폴더를 열어 둔 상태에서 Trigger가 수행될 경우에는 사용 중인 폴더로 폴더가 삭제가 되지 않을 수 있습니다. 열어둔 폴더를 닫으면 다음에 실행 될 때 #max-run-dir 값에 따라 삭제됩니다.

사용 방법

 Note: 사용 중인 Trigger 를 임시 실행 폴더로 설정하기 전 주의사항
 실행 폴더를 사용(#use-run-dir = true) 하여 수행 하기 위해서 Input, Output Data 에 대해 정리가 반드시 필요합니다. 배포(Project) 폴더에서 실행 폴더로 파일이 복사되기에 불필요한 파일을 복사하면서 시간 지연을 유발할 수 있기 에 미리 삭제가 되어야 합니다. 기존에 사용하던 배포(Project) 폴더에서 Experiment 1 폴더를 반드시 삭제하기 바랍니다.
 또한 고정적으로 사용되는 Input Data 가 있다면 이를 제외하고 모두 삭제해 야 합니다. 실행 폴더로 복사 후 수행하기 때문에 사용하지 않는 Input Data 는 모델 폴더에서 정리되어 복사 시간이 오래 걸리지 않도록 정리하여 주시기 바랍니다. 만약, MMC 의 Project > Files 메뉴를 통해 Input Data 를 추가한 경

우에는 반드시 MMC 의 Project > Files 메뉴에서 삭제하여 주시기 바랍니다.

• 116.2 버전부터 임시 실행 폴더에 (base) 폴더가 삭제되었습니다. 116.2 버전 이후 버전으로 업그레이드 시 먼저 (base) 폴더를 삭제 후 업그레이드 하기 바랍니다.

실행 폴더를 이용하여 동일한 Trigger를 병렬적으로 수행하기 위한 Job/Trigger Argument 설정방법은 다음과 같습니다. Job/Trigger에서 설정방법은 동일합니다.

임시 실행 폴더 설정 방법

1. 병렬 수행을 설정할 Job 또는 Trigger을 선택하여 편집 창을 엽니다.

2. Edit Trigger 화면의 Target Job 탭의 #use-run-dir Argument를 체크하여 실행 폴더 사용하 도록 합니다.

3. Edit Job 화면의 Target Job 탭의 #max-run-dir Argument에 허용할 실행 폴더의 수를 입력 합니다.

E See	dit Trigger		– – ×	(
Sche	dule Target J	b Failure Action		
Tar	get job:	MainSim.Sim	~	
You	must specify wha	arguments this task will use.	Reset to defaults	
	z - extended	roperties	A	
#0	daction-excludes			
#0	daction-includes		EqpPlan	
#0	daction-excludes/	n		
#daction-includes/in				
#5	start-time.AdjustM	linutes	0	
#0	use-run-dir		✓	
#r	max-run-dir		2	
#0	overwrite-result			
#0	use-database		\checkmark	
#9	save-database		\checkmark	

실행 폴더 생성 및 수행 절차

Job/Trigger Argument에서 #use-run-dir 값을 true로 설정한 경우에는 모델 폴더에서 실행 폴 더로 model 파일과 Input 폴더가 모두 복사가 됩니다. 복사가 끝난 후 실행 폴더에서 엔진이 수행하게 되며, 실행 폴더 안에서 Job/Trigger Argument 에 설정된 값에 따라 Input/Output DataItem 의 Download 및 Save 에 대한 기능이 동작하게 됩니다.



(i) Note

Input Data의 용량이 큰 경우에는 복사하는데 시간이 걸릴 수 있습니다. Input Data 중 사용하지 않는 Input Data 는 모델 폴더에서 정리하여 복사 시간이 오래 걸리지 않도록 관리하여 주시기 바랍니다.

Dependent Trigger에서 Reference Trigger의 임시 실행 폴더 사용 방법

Dependent Trigger는 경우에 따라 자신이 참조하는 선행 Trigger의 임시 실행폴더를 사용할 필요가 있습니다. 예를 들어 선행 Trigger에서는 Backward/Forward Planning만 수행하고 후 속 Trigger에서 그 결과를 DB에 저장만 한다면 후속 Trigger는 자신의 모델폴더나 임시 실행 폴더가 아닌 선행 Trigger의 임시 실행폴더를 사용하여 수행이 되어야 합니다. #use-parentpath Argument는 Dependent Trigger가 선행 Trigger의 임시 실행폴더를 사용하여 실행할지 여부를 결정하는 Argument이며 MMC에서 설정 방법은 아래 그림과 같습니다.

🔡 Edit Job		23
Basic Parameters		
z - extendedProperties		-
#daction-excludes		
#daction-includes		
#daction-excludes/in		
#daction-includes/in		
#start-time.AdjustMinutes	0	
#use-run-dir	\checkmark	
#max-run-dir	5	
#overwrite-result		
#use-database	\checkmark	
#save-database		
#model-file	Projects\LocaSim\MyModel.vmodel	
#model-dll	Projects\LocaSim\Site.LSE_Planning7.dll	
#model-config		
#experiment	Experiment 1	
#db-to-file		
#file-to-db		
#use-parent-path	\checkmark	
#db-includes		
#db-excludes		-

앞서 기술한 것과 같이 #use-parent-path는 Dependent Trigger에서만 사용 가능한 Argument 이며 주로 선행 Trigger의 결과를 DB에 저장만 하는 #file-to-db = true인 경우에서만 주로 사 용됩니다. 다음은 #use-parent-path Argument의 사용 시나리오 예제입니다.

- Trigger A : Main Trigger
- Trigger B : Trigger A 수행 결과 중 중요 결과를 저장하기 위한 Dependent Trigger(Trigger A 종료 후 수행)
- Trigger C : Trigger A 수행 결과 중 모니터링 결과를 저장하기 위한 Dependent Trigger(Trigger B 종료 후 수행)
- Trigger A 에서 실행 폴더를 사용하도록 체크합니다.(#use-run-dir = true)
- Trigger B 는 Trigger A 가 종료 후 Trigger A 의 결과를 DB 에 저장하도록 설정하고(#fileto-db = true) Trigger A 의 위치를 참조할 수 있도록 설정합니다.(#use-parent-path = true, #use-run-dir = true)
- Trigger C 도 Trigger B 가 종료 후 Trigger A 의 결과를 DB 에 저장하도록 설정하고(#fileto-db = true) Trigger A 의 위치를 참조할 수 있도록 설정합니다.(#use-parent-path = true, #use-run-dir = true)
- Trigger A 종료 후 결과가 Trigger B, Trigger C 에서 저장되는지 확인합니다.

별첨

다음은 #use-run-dir, #max-run-dir, #use-parent-path Argument 사용 시 같이 사용 할 수 없거 나 사용에 주의가 필요한 Argument들에 대한 목록입니다.

- #db-to-file: 임시 실행 폴더가 아닌 Model 폴더에서 수행됩니다.(기존과 동일함)
- file to db job: 임시 실행 폴더에서 수행 그러나 #use-parent-path = true인 경우 새로운 실 행 폴더를 생성하지 않고 선행 Trigger의 실행 폴더를 사용하여 수행합니다.
- overwrite job: 운영시 #use-run-dir 과 함께 실행 할 수 없는 기능입니다. 만약 함께 수행 시 가장 마지막에 수행한 폴더에서 수행되며, 실행 폴더가 추가 생성되지 않습니다.

Trigger를 특정 **Mozart Server DLL**에서 실행 하는 방법

기능설명

Mozart Server 실행시 특정 버전으로 실행 할 수 있는 기능입니다. 즉, Mozart Server 버전 업 그레이드 후 업그레이드 되기 전의 기존 모델들을 돌릴 수 있는 기능입니다.

기능 추가 배경

운영 환경은 안정성이 반드시 보장되어야 합니다. Mozart Server의 버전이 업그레이드가 되 면 그 안에서 운영되고 있는 모델과 모델(엔진) DLL도 업그레이드된 버전의 Mozart Server와 호환이 가능하도록 다시 빌드 및 배포가 되어야 합니다. 만약, 예기치 못한 상황에서 새로 빌 드된 모델에서 오류가 발생하면 문제 해결 전에는 이전에 안정적으로 운영된 버전으로 롤백 을해야합니다. 여기서 롤백은 모델 및 모델(엔진) DLL 뿐만 아니라, Mozart Server도 이전 버 전으로 되돌리는 것을 의미합니다. 버전 롤백의 경우 많은 시간이 소모되며 그 과정에서 Trigger를 실행하는 스케쥴에도 영향을 끼칠 수 있습니다.

그래서 운영 안정성을 보장하면서 롤백에 의한 Trigger 실행 주기 영향 등의 문제를 예방하기 위해 MMC에서 Trigger 실행을 호스팅 할 Mozart Server를 버전을 지정 할 수 있는 기능들이 있습니다. 해당 기능을 통해 Mozart Server 버전 업그레이드 후 Trigger 실행에 실패하더라도 롤백 과정 없이 이전에 안정적으로 운영된 버전으로 Trigger를 실행 할 수 있습니다.

기능 구성

상이한 버전의 서버를 실행시키기 위한 Argument는 두 가지가 있습니다. 하나는 #host-dir 그 리고 또 하나는 #host-version입니다. 두 Argument의 동작방식은 동일하지만 Trigger 실행이 참조하는 위치는 다릅니다. 차이는 다음과 같습니다.

Argument DataType Description

#host- version	string	MOZART Configuration for Server에서 사용자가 지정한 Domain Library 설치 경로. 입력값은 제품의 버전 정보입니다.
#host-dir	string	상대경로: WorkingDir 내 서버 폴더 위치 절대경로: 사용자가 직접 입력한 경로

사용방법 및 예시

#host-dir

#host-dir 사용방법은 다음과 같으며 과정 중 사전 준비 작업이 포함되어 있습니다. 이 방법은 설치된 Mozart Server가 116.0 미만인 경우에만 사용하시기 바랍니다.

1. Mozart Server의 버전 업데이트를 하기 앞서, 현재 버전의 Mozart Server 폴더를 Working Directory 또는 임의 경로에 복사 합니다. Mozart Server 버전이 116.0 미만인 경우 보통 *C:\Program Files (x86)\VMS\Mozart\Server\Bin*에 설치되어 있습니다. 아래의 샘플은 Mozart Server를 Working Directory에 복사하면서 Bin 폴더의 이름을 버전명으로 변경했습니 다.

√ → → → → → → → → → → → → → → → → →		😂 🖪 🗈
Name	Date	Size
2018.2.113.0	2020-01-16 18:35:03	
2018.3.114.1	2020-01-16 18:35:12	
2019.115.1	2020-01-16 18:35:12	
2019.116.0.61	2020-01-16 18:35:21	
2019.116.1.28	2020-01-16 18:34:59	
Changesets	2020-03-09 19:56:10	_
Execution	2020-03-25 13:38:39	
Logs	2020-03-25 11:07:23	
Projects	2020-03-09 19:56:10	
Results	2020-03-09 19:56:10	

2. Working Directory에 백업이 완료되었으면, Mozart Server의 버전을 업그레이드 합니다. #host-dir Argument는 Mozart Server 및 MMC 115.0 버전 이상부터 사용이 가능합니다. 3. Mozart Server 및 MMC의 업데이트가 완료되었으면 Mozart Client가 설치된 경로(i.e *C:\Program Files (x86)\VMS\Mozart\Client\Bin*)에서 **MozartManagementConsole2.exe**를 실행합니다.

4. Server Explorer에서 접속 할 서버를 선택, 마우스 우 클릭하여 **[Connect Server]**를 선택합 니다.

5. 서버 접속에 필요한 정보(ID/PW) 입력 후 서버에 접속합니다. 다음, Server Explorer에서 Triggers를 더블 클릭 또는 마우스 우 클릭, **[Open]**을 선택하여 엽니다.

6. 활성화된 Triggers 탭에서 호스팅 서버를 변경하고자 하는 Trigger를 선택하여 Trigger 편집 다이얼로그를 엽니다.

7. **[Target Job]** 탭으로 이동하여 **[z - extendedProperties]** 카테고리의 목록 마지막까지 내려 간 후 #host-dir Argument에 Working Directory에 백업해둔 Mozart Server의 폴더명을 입력합 니다. 아래는 예제입니다.

🗱 Edit Trigger	-	
Schedule Target Job Failure Action		
Target job: ServerTest.ServerTest	~	
You must specify what arguments this task will use.	Reset to	
#use-parent-path		
#db-includes		
#db-excludes		
#zip	\checkmark	
#zip.Path	ServerTest	
#zip.FileNamePostfix	ServerTest1.Oracle	
#zip.FileNameTemplate		
#zip.UpdateToRecent		
#more-runs	0	
#run-index		
#log-level		
#log-dir	ServerTest	
#dataSource-set-default	K1=V1;K2=V2	
#datasource-set-default-exception		
#performance-profiling	✓	
#host-dir	2018.3.114.1	•••
#host-version		
		T
In and a Record 33 of 34 ► IN IN + A 4 X 4		+
Schedule Now	OK	Cancel
ocheddie 140W	OK	Cancer

위와 같이 설정하면 해당 Trigger는 현재 설치된 Mozart Server 버전과 관계없이 #host-dir에 지정된 폴더 안에 있는 Mozart Server 및 Domain Library DLL를 참조해서 실행합니다. 단, #host-dir에 설정된 DLL를 참조해서 Trigger를 실행하기 위해서는 대상 모델 및 엔진 DLL 파 일이 #host-dir의 버전과 항상 동기화되어 있어야 합니다. 위의 예제에서와 같이 #host-dir로 지정된 폴더의 DLL들이 2018.3.114.1이면, Trigger가 실행하는 모델 및 엔진 DLL 파일은 2018.3.114.1로 빌드되어 있어야 합니다.

#host-version

host-version 을 사용하는 방법은 다음과 같습니다. #host-version Argument는 116.0 버전 이 상의 Mozart Server 및 MMC에서 사용이 가능합니다.

1. **MozartManagementConsole2.exe** 를 실행합니다. 실행 파일은 Mozart Client 가 설치된 폴더에 있습니다. (i.e *C:\Program Files (x86)\VMS\Mozart\Client\Bin*) 4. Server Explorer에서 접속 할 서버를 선택, 마우스 우 클릭하여 **[Connect Server]**를 선택합 니다.

5. 서버 접속에 필요한 정보(ID/PW) 입력 후 서버에 접속합니다. 다음, Server Explorer에서 Triggers를 더블 클릭 또는 마우스 우 클릭, **[Open]**을 선택하여 엽니다.

6. 활성화된 Triggers 탭에서 호스팅 서버를 변경하고자 하는 Trigger를 선택하여 Trigger 편집 다이얼로그를 엽니다.

7. **[Target Job]** 탭으로 이동하여 **[z - extendedProperties]** 카테고리의 목록 마지막까지 내려 간 후 #host-version Argument에 Execution 폴더(i.e C:\Program Files (x86)\VMS\Mozart\Server\Execution에 있는 버전 중 대상 Trigger를 호스팅하고자 하는 버 전을 입력합니다. 아래는 예제입니다.

Local Disk (D:) > Program Files > VMS	» Mozart » Server » Ex	ecution
Name	Date modified	Туре
2019.115.000.0	7/25/2019 2:33 PM	File folder
2019.115.100.0	7/24/2019 2:44 PM	File folder
2019.116.000.16	7/24/2019 2:44 PM	File folder

🗱 Edit Trigger		_		×
Schedule Target Job Failure Action				
Target job: ServerTest.ServerTest	~			
You must specify what arguments this task will use.	Reset to			
#use-parent-path				
#db-includes				
#db-excludes				
#zip	\checkmark			
#zip.Path	ServerTest			
#zip.FileNamePostfix	ServerTest1.Oracle			
#zip.FileNameTemplate				
#zip.UpdateToRecent				
#more-runs	0			
#run-index				
#log-level				
#log-dir	ServerTest			
#dataSource-set-default	K1=V1;K2=V2			
#datasource-set-default-exception				
#performance-profiling	v			
#host-dir				
#host-version	2019.115.100.0			
H4 44 4 Record 34 of 34 → H+ + + √ × 4				▼
Schedule Now		ОК	Cance	el

i Note:

Mozart Server의 Execution 폴더는 Mozart Server Configurator에서 지정 할 수 있 습니다. Execution 내 버전 별 폴더는 해당 버전의 Domain Library mdz 패키지 파 일을 설치하면 자동 생성됩니다. #host-version은 Mozart Server Configurator에서 지정한 Execution 폴더에서 Argument에 입력한 버전명의 폴더를 찾습니다.

실행 우선 순위

- 1. #host-version
- 2. #host-dir (1순위 사용자 지정 경로, 2순위 WorkingDirectory)

3. 1,2 값이 없을 경우 Mozart Server 설치 된 Execution 폴더 내 가장 높은 버전명을 가진 폴

더(수동으로 폴더를 추가할 경우 버전명 규칙에 맞게 폴더명 생성이 필요합니다.)

만약 아래 그림과 같이 #host-dir 과 #host-version 의 값이 각각 존재할 경우에는 2019.115.100.0 버전이 실행되게 됩니다.

🗱 Edit Trigger	— C	X C
Schedule Target Job Failure Action		
Target job: MainSim.Sim	~	
You must specify what arguments this task will use.	Reset to defaults	
#file-to-db		
#db-includes		
#db-excludes		
#zip	\checkmark	
#zip.Path	MainSim	
#zip.FileNamePostfix		
#zip.FileNameTemplate		
#zip.UpdateToRecent		
#more-runs	0	
#run-index		
#log-level		
#log-dir	MainSim	
#dataSource-set-default	K1=V1;K2=V2	
#datasource-set-default-exception		
#performance-profiling	✓	
#host-dir (2)	Server1141	
#host-version (1)	2019.115.100.0	
H4 44 4 Record 1 of 34 → H+ H+ + √ × 4		* *
Schedule Now	ОК С	ancel

Dependent Trigger 예제

Dependent Trigger 의 한 유형으로 대상이 되는 Trigger 에서 수행한 Job 의 결과값을 확인하 여 실행 여부를 설정하는 예를 통해 설정 방식을 설명합니다. 타 시스템과의 연계 실행을 위 해 특정 DB 의 Table 을 사용하는 경우가 이러한 Dependent Trigger 를 사용할 수 있는 대표 적인 예라고 할 수 있습니다.

아래 그림과 같이 특정 시스템에서 작업을 실행 한 후, 결과를 확인하고 우리가 수행할 Job 을 실행해야 하는 경우 1) 결과 확인을 위한 Job 을 생성 하고 타 시스템의 결과를 2) 모니터 링하도록 Trigger 를 설정 합니다. 그리고 마지막으로 과정 2에서 설정한 Trigger 를 대상으로 3) Dependent Trigger 를 만듭니다.

Schedule Target Job Failure Action Trigger name: Monitoring Category:	🗱 New Trigger		_		×
Trigger name: Monitoring Category:	Schedule Target	: Job Failure Action			
Settings One Time Start: 2018-12-11 ♥ ♥ \$4:00:00 ♥ Set to now Simple Expire: 2018-12-12 ♥ ♥ \$3:53:00 ♥ Daily Weekly Becur every: 00:10:00 intervals Per years Per years	Trigger name: Category:	Monitoring			
O Dependent Repeat count: -1 ⊕ (* -1: infinitely)	Settings One Time Simple Daily Weekly Monthly Dependent	Start: 2018-12-11 □▼ ♀₹ 4:00:00 ▼ Set to now □ Expire: 2018-12-12 □▼ ♀₹ 3:53:00 ▼ Recur every: .00:10:00 intervals Repeat count: -1 (* -1: infinitely)			
Advanced Settings Priority: 0 ÷ Stop task if it runs longer than: 00:00:00 Retry Count: 0 ÷ Enabled 7race Schedule Now OK	Advanced Setting Priority: Stop task if it Enabled Trace Schedule Now	0 0 Retry Interval: 0 Retry Count: 0	00:	00:00 0÷	

실행결과를 DB에 업데이트 하는 로직 추가(Monitoring 대상 Model)

Mozart Project 의 실행이 정상적으로 종료되었는지 여부를 외부의 DB 로 기록하여 Monitoring 하기 위해 아래의 절차를 사용합니다.

1. Output 결과 기록을 위한 Output Data Item을 정의합니다.

Mfg	JМс	del/Outputs/itor	ing/ResultFlag 🗙 Persist Col	nfig/Ou	tput Co	nfig		
	In	heritance						
	D	escription						
0	Pr	operties 🚺						
		Name	PropertyType	Key	Null	Editor		Са
		STATE	string		V		•	
		STATE_TIME	datetime		V		•	
		EXCEPTION	string		1		•	
Þ	*	P4	string 🚽		1		•	
			F					

2. Output Persist Config에 Monitoring table 속성에 결과 기록을 위한 Output data item(1번 항목에서 정의한 item)을 지정합니다.

Persist Config/Output Config ×	
Add - Remove	
⊡ Outputs	Name: Outputs
Pegging Persists	Model: * ~
	☑ Log performance
	Thread count: 3
	DB job retry count: 3
	Exception Policy
	StopAtThrown ~
	Monitoring table
	output.MonitorTable

3. MainControl의 ShutDown FEAction을 구현합니다. 아래는 예제 소스입니다.



4. 결과 Ouput에 대한 DataAction 설정을 통해 Monitoring용 DB에 결과를 기록합니다.



실행결과 확인을 위한 Job 생성(Monitoring Task)

실행결과 확인을 위한 Job 은 기본적으로 **1) 특정 DB로 부터 데이터를 조회하여 결과를 확인** 하고 **2) Target Job 실행여부를 기록**한 후 **3) 확인된 결과를 반환**하도록 작성합니다.

Dependent Trigger 의 **ReturnIfTrue** Type Job 의 return 값은 Main 함수의 **ModelContext**의 **Result** 값에 설정합니다. 설정값은 **boolean** 형식입니다.

대상 Job의 실행 결과를 확인하기 위한 Monitoring Task 준비는 아래의 절차를 사용합니다.

1. Monitoring 대상 Job의 실행 결과 Output을 기록하는 DB table를 조회하기 위한 Input Data Item을 정의합니다.

)	Inheritance						
)	Description						
	Properties 1		1]/4 🕨	• ▶ ♣ ×		
	Name	PropertyType	Key	Null	Editor	Hidder	Caption
E.	VERSION_NO	string			~		
	STATE_TIME	DateTime			~		
	EXCEPTION	string			~		
	STATE	string			~		
					~		25 25

2. 새롭게 정의한 Data Item에 Default Data Action에는 DB로부터 Job 실행 결과를 조회 할 수 있는 쿼리, 그리고 Data Action을 하나 더 추가하여 Monitoring Task에서 업데이트 할 쿼 리를 정의 합니다. 이때 Select문을 실행 할 DataAction Active 로 설정합니다.

DataAction 추가 화



Select문 구현 예제

	+ - + +	CommandType	Text ~	DataSource	-	~	Bind Table		
⊡- Select L Cmd1		SELECT A.VER A.STA A.EXC A.STA FROM MOZ_M	SION_NO, TE_TIME, EPTION, TE ONITOR_TABLE A						
DataSource			6 (
FabDB	~	Name *	Size	DbType	~	Direction	Precision	Scale	SourceVersion

Update문 구현 예제

+ - † 4	CommandType T	ext v D	ataSource -			~	Bind Table	e			
Update Cmd1 Activate	UPDATE MOZ_MONI SET A.VERSIC A.STATE A.EXCEPT A.STATE	TOR_TABLE A NLNO-@VERSION_N IIME-@SIATE_IIM ION-@EXCEPTION, @STATE	10, IE,								
DataSource	100%	0									
FabDB ~	Name	Size	DbType		Direction		Precision	Scale	SourceVers	ion	SourceColumn
	@VERSION_NO	50	AnsiString	~	Input	~	0	0	Current	~	
	@STATE_TIME	0	DateTime	~	Input	~	0	0	Current	~	
	@EXCEPTION	2147483647	AnsiString	~	Input	~	0	0	Current	~	
	@STATE	50	AnsiString	~	Input	~	0	0	Current	~	
	•			~		~				~	

3. 대상 Job 수행 결과에 따라 결과값을 true를 반환하는 로직을 구현합니다. 아래는 [Main > Run] 함수의 구현 예입니다.

```
1 //Main > Run 함수 구현
2 public void MONITORING(ModelContext context, ref bool handled)
3 {
4 var info = InputMart.Instance.MonitorTable.Rows.FirstOrDefault();
5
6 //이전 Job 결과가 기록 그리고 정상 수행된 경우 Monitor Table 업데이트
7 if(info!= null && info.STATE == "SUCCESS")
8 {
9 //다음 Job 성공까지 본 조건문에 들어오지 않게 STATE 값 변경.
```
```
info.STATE = string.Empty;
info.STATE_TIME = DateTime.Now;
//MonitorTable 이력을 DB에 업데이트 하기 위한 쿼리 실행 및 실행 여부를 ?
var dir = Path.Combine(context.ModelDirectory, context.VModelName
var model = Mozart.Task.Model.ModelEngine.Load(dir);
var source = new MonitorTable[] {info};
var result = InputAccessHelper.Save<MonitorTable>(model, "MonitorT
var result = InputAccessHelper.Save<MonitorTable>(model, "MonitorT
//쿼리가 정상 실행된 경우 result = 1이며, 그 경우 결과값을 true 반환
//ReturnIfTrue 타 Dependent Trigger를 실행함.
if(result > 0)
context.Result = true;
}
```

모니터링 Trigger 설정

2번에서 개발한 Monitoring을 위한 vModel 및 dll를 MMC를 통해 Upload하고, Job을 생성하 여 해당 Job을 실행하기 위한 Trigger를 설정합니다. 통상적으로는 Monitoring Job을 수행하 기 위한 Trigger는 Simple Trigger로 설정을 하며 주기는 경우에 따라 최소 1분, 최대 1시간으 로 설정을 합니다. 아래의 예제는 매일 오후 4시부터 10분 단위로 15분간 Trigger가 수행됩니 다. for a duration of 옵션은 Repeat count가 무제한(-1)으로 설정된 경우에만 활성화 됩 니다. 만약 상시로 10분마다 상태를 체크해야 하는 경우에는 for a duration of 옵션을 설 정하지 않습니다.



Dependent Trigger 설정

Monitoring Trigger가까지 등록을 하였다면, 마지막으로 Target Job 실행을 위한 Depdendent Trigger를 생성하여 설정합니다. 이때 Referred Trigger 는 Monitoring을 수행하는 Trigger 를 선택하고, Monitoring 결과값이 true인 경우에만 실행이 필요한 관계로 Execution type 을 ReturnIfTrue으로 선택합니다. 아내는 설정 예제입니다.

🔡 New Trigger		_		×
Schedule Target	Job Failure Action			
Trigger name:	Backup_Simulation			
Description:				
Settings				
One Time	Start: 2018-12-11 💷 오후 4:00:01 文 Set to now			
O Simple	□ Expire: 2018-12-11 □▼ 으후 4:00:01 🗼			
	Referred Trigger: Monitoring ~			
 Dependent 	Execution type:			
	Execution delay: 00:00:03			
Advanced Setting	S			
Priority:	0 🚔 🗌 Retry Interval:	00	:00:00	
Stop task if it r	runs longer than: 00:00:00 Retry Count:		0 🌲	
Enabled				
Trace				
Schedule Now	OK		Cance	

Extended Arguments

MOZART Project 로 생성된 Task(DII 파일) 와 Model 은 실제로 실행될 경우 MOZART 실행엔 진의 Model Task 를 통해 수행됩니다. 이러한 Model Task 의 실행 옵션을 조정하기 위해 사 전 정의된 Arguments 가 System Arguments 입니다. 개발자는 Model 의 Input Arguments 설 정 시 사전 정의된 System Arguments 를 지정함으로써 사전 정의된 실행 옵션을 활용할 수 있습니다. 아래 표는 사전 정의된 System Arguments 에 대한 설명합니다.

Basic Arguments

Argument Name	Argument 설명	Data Type
#experiment	Model 실행 시 결과를 생성할 Experiment 명. 기본값은 "Experiment 1"	string
version-no	Model Version 명 (기본값은 {model-name}-{yyyyMMdd- HHmmss} 형식)	string
model-name	versionNo 없을 때 versionNo에 사용될 이름	string
start-time	Task 시작시간 (Simulation Clock)	DateTime
end-time	Task 종료시각 (Simulation Clock)	DateTime
period	Plan&Schedule 기간	float
period-unit	period 설정 단위(Days, Hours). 기본값은 일단위입니다 (Days).	string
#start- time.AdjustMinutes	Job 실행 시작 시간에서 해당한 시간만큼 조정한 시간 을 start-time으로 설정하도록 조정하는 입력변수	int
#model-file	실행 Model 명을 포함한 Full Path	string
#model-dll	Model 을 실행할 DII 의 Full Path	string

Data Download/Upload 옵션 Arguments

Argument Name	Argument 설명	Data Type
#overwrite_result	Result 덮어쓰기 여부	boolean
#use-database	Database 읽기 여부	boolean
#save-database	Output Data 를 Database 에 쓰기 여부	boolean
#db-to-file	시뮬레이션은 실행하지 않고 Database Sync 만 진행 여부 (기본값 = false)	boolean
#file-to-db	시뮬레이션은 실행하지 않고 Output에 대해 Database Sync만 진행 여부 (기본값 = false), True인 경우 Result가 생성되지 않음	boolean
#db-includes	Input Data Sync 대상 Table List 파일 이름, 대 상 Table 들만 Sync 진행	string
#db-excludes	Input Data Sync 제외 Table List 파일 이름, 제 외 Table 들을 제외한 모든 Table Sync 진행, #daction_includes 동일 값이 있는 경우, excludes 반영 안됨	string
#daction_excludes	시뮬레이션 완료 후 DataAction 실행하지 않 을 Table List, Comma 로 구분	string
#daction_includes	시뮬레이션 완료 후 DataAction 실행할 Table List, Comma 로 구분	string
#daction_excludes/in	시뮬레이션 시작 시 Input Schema 의 DataAction 중 실행에서 제외할 Action List, Comma 로 구분	string

#daction_includes/in	시뮬레이션 시작 시 Input Schema 의 DataAction 중 실행할 Action List, Comma 로 구분	string
#dataSource-set- default	모델에서 기본으로 사용할 Connection String 설정. Key 는 DataSource의 이름이고, Value는 Connection String의 이름이며, 여러 Key/Value 입력 시 Semicolon(;)로 구분	Dictionary <string, string></string,
#datasource-set- default-exception	#dataSource-set-default의 Key/Value에 해당 되는 Connection String을 못 찾 시 Exception 발생 여부	boolean
#commandTimeout	DataAction의 Command 별 Timeout을 설정. 시간 단위는 초 단위이며 기본값은 180초(3 분)	integer

Logging/Performance Arguments

Argument Name	Argument 설명	Data Type
#log-dir	Log 출력 경로, Working Directory\Logs 기준의 상대경로	string
#log-level	로그 기록 수준 설정 Verbose~Fatal 설정 가능	string
#performance- profiling	Model 수행의 성능 집계 여부 (기본값 = false), false인 경우 Triggers,Monitoring에서 Trigger Execution Log 정보가 조회되 지 않으며, Performance에서도 정보가 보이지 않음	boolean

Run Arguments

Name	Argument 설명	Data Type
#more- runs	Model 의 반복 실행 횟수	int
#more- config- [runindex]	반복 실행 회차 별 Arguments 값을 설정하는 변수지정되지 않은 경 우 이전 회차의 Arguments 값을 그대로 사용하며, MMC 에서 자동으 로 생성	string
#run-index	반복 실행 회차 정보이며, MMC 에서 자동으로 생성	int
#exec- dependent- list	하나의 Trigger에 여러 Dependent Trigger가 연결되어 있는 경우 이를 선택적으로 실행 할 수 있는 Argument. 실행 할 Trigger 명을 입력하 며 하나 이상 추가 쉼표(,)로 구분. #exec-dependent-list에 추가되는 Depdenet Trigger의 경우 Execution Type을 ReturnIfTrue로 설정해야 함. 자세한 내용은 여기를 참고	string

Temp Folder Run Arguments

Argument Name	Argument 설명	Data Type
#use-run- dir	Trigger 실행 시 임시실행 폴더 생성 여부. 자세한 내용은 여기를 참 고 바랍니다.	boolean
#max-run- dir	임시실행 폴더 생성 개수 설정. Max 값을 초과하는 경우 가장 마지 막에 생성된 폴더가 삭제	int
#use- parent- path	Dependent Trigger 실행 시 임시실행 폴더를 생성하지 않고 Reference Trigger의 가장 최근 임시실행 폴더를 사용할지 여부. 현 재는 #file-to-db에서만 사용을 권	boolean

Zip Model Arguments

Model 실행 후 압축파일로 Model 을 압축할때 사용하는 규칙에 대한 설정 Arguments 입니다.

Argument Name	Argument 설명	
#zip	시뮬레이션 완료 후 Model 압축 여부	boolean
#zip.FileNameTempate	압축 파일 이름 저장을 위한 Template기본 Template 은 "\${model_name}_\${zip_now}\${zip_postfix}"사용가 능한 keyword 는 아래와 같습니다.\${model_name}: Model 명\${now}: 압축실행 시점 시각 (DateTime)\${zip_now}: 압축하는 시각문자열(format : yyyyMMddHHmmss)\${zip_postfix}: 설정한 압축파 일 이름 접미사\${version_no}: Model 실행 Version 명	string
#zip.FileNamePostfix	압축 파일 이름 접미사	string
		3
#zip.Path	압축파일 생성 경로, 정하지 않으면 Model 파일이 있 는 위치에 생성됨MOZART Server 의 Working Directory 기준으로 상대경로로 설정하며, 설정하지 않 는 경우 Working Directory 에 생성됨	string

Hosting Arguments

Job/Trigger 실행을 Hosting 할 서버 경로를 설정하는 Argument입니다.

Argument Name	Argument 설명	Data Type

#host-dir	등록된 Job/Trigger의 Task DLL 및 Model이 Server 머신에 설치된 버 전이 아닌 다른 버전을 통해 Hosting되어 수행이 필요 할 시 Hosting 할 Server DLL이 위치한 경로를 지정합니다. 해당 Argument 사용 시 Hosting 대상 Server DLL 파일들이 WorkingDirectory 내 존재해야 합 니다. 자세한 내용은 여기를 보시기 바랍니다.	string
#host- version	등록된 Job/Trigger의 Task DLL 및 Model이 Server 머신에 설치된 버 전이 아닌 다른 버전을 통해 Hosting되어 수행이 필요 할 시 Hosting 할 Server DLL의 버전을 지정압합니다. 해당 Argument는 Mozart Server가 설치된 경로의 Execution 폴더 내 버전을 의미하며, 동작 방 식은 #host-dir과 동일합니다. 만약 #host-dir과 #host-version이 동시 에 설정이 되어 있으면, #host-version에 설정된 값이 우선순위 높습니 다. 자세한 내용은 여기를 보시기 바랍니다.	string

Log 조회

MMC2에서 Trigger에 대한 Task와 Persist 로그는 기본적으로 Job/Trigger에 연결된 Project 의 Logs 폴더를 통해 조회가 가능합니다. 또한 Triggers 및 Monitoring에서도 Trigger 수행 로 그 조회가 가능합니다.

Projects에서 로그 조회 하기

- 1. Server Explorer -> Projects 노드에서 Trigger와 연결된 Project 노드를 엽니다.
- 2. Logs 폴더를 더블 클릭 또는 오른쪽 마우슨 버튼 메뉴를 통해 [Open] 을 클릭합니다.



MMC2_TEST/Projects/Colla_Test/Logs ×								
🔇 - 🕥 - 📄 📷 workingDirectory#Logs#Colla_Test# 💈 🚺 🚹								
Name	Date	Size	Attributes					
task-Fab6-20170323-170815.log	2017-03-23 17:08:24	4 kb	-a					
task-Fab6-20170323-174612.log	2017-03-23 17:46:19	4 kb	-a					
task-Fab6-20170323-175109.log	2017-03-23 17:51:16	4 kb	-a					
task-Fab6-20170327-184003.log	2017-03-27 18:40:09	9 kb	-a					
task-Fab6-20170329-091346.log	2017-03-29 09:13:53	4 kb	-a					
task-Fab6-20170329-091438.log	2017-03-29 09:14:48	5 kb	-a					
task-Fab6-20170329-092246.log	2017-03-29 09:22:53	4 kb	-a					

MOZART Working Directory 하위의 Logs 폴더에 기본적인 Task 수행 로그를 기록합니다. Logs 의 System 폴더에는 Server 의 전체 실행 Log 를 기록하며, Model 에 별도의 Config 파 일을 지정(Log 지정방법 참고)하여 Log 파일 경로를 지정하지 않은 경우 Log 파일은 Logs 폴 더에 기록됩니다. 3. 대상 시간대의 Task Log 파일을 더블 클릭하여 실행 Log 결과를 확인합니다.

Trigger Log 는 실행 시점에 2개의 Log 를 남기며, Log 파일 명에는 Model 의 Arguments 중 model-name 에 입력된 Parameters 값을 포함합니다.

 task Log : Task Module 실행 시 Log 결과를 기록하는 Log 파일입니다. model-name 을 Input Arguments 에 등록하고 설정한 경우 해당 model-name 을 포함하여 Log 파일이 생 성됩니다. 예를 들어 model-name 을 'SAMPLE' 로 설정한 경우, Log 파일 명은 task-SAMPLE-20150101-125648.log 와 같이 생성됩니다. Log 파일을 더블 클릭하면 기본 Editor 로 해당 Log 파일을 열어 조회할 수 있습니다.





• persists Log : Task Module 수행 전/후의 persist 처리 결과를 기록하는 Log 파일

Triggers에서 로그 조회하기

1. Server Explorer -> Triggers 노드를 더블 클릭 또는 마우스 오른쪽 버튼 메뉴 **[Open]** 을 클릭 합니다.

2. Triggers Tab이 열리면 Log를 조회하고자 하는 Trigger를 선택합니다.

3. 상단 메뉴바의 **[Open Log Folder]** 를 클릭하면 해당 Trigger의 Log를 조회 할 수 있는 탭이 활성화 됩니다.

4. 조회하고자 하는 로그 파일 더블 클릭 시 메모장에서 열립니다.

Monitoring에서 로그 조회하기

1. Server Explorer -> Monitors 노드에서 Log를 조회하고 싶은 View를 더블 클릭 혹은 마우스 오른쪽 버튼 메뉴 **[Open]** 을 클릭합니다.

2. Triggers Tab에서와 마찬가지로 Log를 조회하고자 하는 Trigger를 선택 후 상단 메뉴바의 [Open Log Folder] 를 클릭합니다.

3. 조회하고자 하는 로그 파일 더블 클릭 시 메모장에서 열립니다.

System Log 조회

1. System Log 폴더는 기존 MMC와 동일한 구조를 가지고 있으며, MMC를 설치 시 System Log 조회 항목이 자동 생성됩니다.

2. app.log 파일이 조회일의 Log 이며, 과거 일자의 Log 는 일자별 Log 파일을 찾아 조회합니다. 익일이 되면 app.log 파일을 일자별 파일로 백업하고 app.log 에는 당일 Log 만 남기게 됩니다.



Monitoring 사용법

Monitoring에서는 MMC에 등록된 Trigger의 수행 상태, Trigger를 수행하는 대상 서버의 CPU/Memory 사용 현황, Trigger 별 로그 조회 및 Task의 단계 별 수행 소모 시간 등을 확인 할 수 있습니다.

MOZART Server를 설치하게 될 시 Server Explorer에서 대상 서버에 Monitor 노드가 생성이 되며 기본적으로 Trigger들의 가장 최근 상태를 확인 할 수 있는 All View 및 오류 발생 이력을 확인 할 수 있는 Error View가 Child 노드로 생성이 됩니다. 특정 Trigger에 대한 Monitoring을 하기 위해서는 사용자가 직접 View를 추가 할 수 있으며 이때 사용자가 추가한 View는 해당 Trigger의 수행 별 Monitoring 이력을 확인 할 수 있습니다. 다음은 Monitoring의 기본 인터페 이스 및 기능에 대해 기술합니다.

노드 설명

Server Explorer 상에서 Monitor 노드에 대한 설명입니다.



- All: 등록된 Trigger가 단 1회라도 수행이 되는 경우 All View에 이력이 남습니다. All에서 는 대상 Trigger가 가장 최근에 수행된 정보만 표시됩니다. 즉 A라는 Trigger가 9시에 한 번 수행이 되고 10시 또 수행이 되었을 시 All에서는 A Trigger에 대해 10시에 수행된 정 보만 표시가 됩니다.
- Errors : 대상 Trigger의 수행 오류 이력 정보를 보여줍니다. All에서는 대상 Trigger의 가 장 최신 정보만을 보여주는 반면 Errors에서는 대상 Trigger가 오류가 발생 할 때 마다 기 록이 됩니다. 이에 Errors에서는 사용자가 기록을 조회하고자 하는 기간을 설정을 할 수 있습니다.

A	ito Refresh Interval: No	ormal (30s) 🔹							Search Option:	30 days Que	ry
D	rag a column header he	re to group by that colu	mn								
	TriggerName	Scheduled	Start	End	Elapse	Status	Result	Message			
9											*
•	Load Test_Monitor	2017-05-02 19:23:16	2017-05-02 19:57:09	2017-05-02 19:58:10	00:01:01	Complete	FAIL	System.IO.IOException: 'e:\#vms\#mozart			
	LoadTest_Monitor	2017-05-02 19:11:16	2017-05-02 19:55:48	2017-05-02 19:56:40	00:00:52	Complete	FAIL	System.IO.IOException: 'e:\vms\vmsozart			
	LoadTest_Monitor	2017-05-02 18:56:16	2017-05-02 19:53:22	2017-05-02 19:54:11	00:00:49	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:53:16	2017-05-02 19:53:21	2017-05-02 19:54:08	00:00:47	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:48:16	2017-05-02 19:53:10	2017-05-02 19:53:27	00:00:17	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:47:16	2017-05-02 19:52:13	2017-05-02 19:53:16	00:01:03	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:46:16	2017-05-02 19:52:09	2017-05-02 19:53:16	00:01:07	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:45:16	2017-05-02 19:52:05	2017-05-02 19:53:16	00:01:11	Complete	FAIL	System.IO.IOException: 'e:\www.wmozart			
	LoadTest_Monitor	2017-05-02 18:44:16	2017-05-02 19:52:04	2017-05-02 19:53:16	00:01:12	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:43:16	2017-05-02 19:40:05	2017-05-02 19:53:16	00:13:11	Complete	FAIL	System.IO.IOException: 'e:\vms\vmsvmozart			
	LoadTest_Monitor	2017-05-02 18:41:16	2017-05-02 19:40:03	2017-05-02 19:53:16	00:13:13	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:40:16	2017-05-02 19:40:02	2017-05-02 19:53:16	00:13:14	Complete	FAIL	System.IO.IOException: 'e:\vms\vmsvmozart			
	LoadTest_Monitor	2017-05-02 18:38:16	2017-05-02 19:39:27	2017-05-02 19:53:16	00:13:49	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			
	LoadTest_Monitor	2017-05-02 18:39:16	2017-05-02 19:39:27	2017-05-02 19:53:16	00:13:49	Complete	FAIL	System.IO.IOException: 'e:\#vms\#mozart			
	LoadTest_Monitor	2017-05-02 18:33:16	2017-05-02 19:39:26	2017-05-02 19:53:16	00:13:50	Complete	FAIL	System.IO.IOException: 'e:\vms\vmsvmozart			
	LoadTest_Monitor	2017-05-02 18:34:16	2017-05-02 19:39:26	2017-05-02 19:53:16	00:13:50	Complete	FAIL	System.IO.IOException: 'e:\#vms\#mozart			
	LoadTest_Monitor	2017-05-02 18:36:16	2017-05-02 19:39:26	2017-05-02 19:53:16	00:13:50	Complete	FAIL	System.IO.IOException: 'e:\vms\vmsvmozart			
	LoadTest_Monitor	2017-05-02 18:32:16	2017-05-02 18:42:12	2017-05-02 19:38:27	00:56:15	Complete	FAIL	System.IO.IOException: 'e:\#vms\#mozart			
	LoadTest_Monitor	2017-05-02 18:29:16	2017-05-02 18:42:11	2017-05-02 19:38:27	00:56:16	Complete	FAIL	System.IO.IOException: 'e:\\vms\\mozart			
	LoadTest_Monitor	2017-05-02 18:27:16	2017-05-02 18:42:10	2017-05-02 19:38:27	00:56:17	Complete	FAIL	System.IO.IOException: 'e:\#vms\#mozart			
	LoadTest_Monitor	2017-05-02 18:24:16	2017-05-02 18:42:02	2017-05-02 19:38:27	00:56:25	Complete	FAIL	System.IO.IOException: 'e:\vms\vmszart			÷

 사용자 정의 View : 사용자가 추가한 View입니다. 사용자 View에서는 사용자가 선택한 Trigger에 대한 모든 수행 이력의 Monitoring 상태를 확인 가능합니다. 페이징 기법이 적 용이 되어 활성화된 창 크기에 따라 10개 단위로 리스트가 출력 됩니다. 사용자 View 추 가 관련해서는 Monitoring View 등록/수정 방법을 참고 바랍니다.

Monitoring Table 설명

다음은 Monitoring 화면에서 Main이 되는 Trigger의 상태 정보를 표시하는 Table에 대한 설명입니다.

- **TriggerName** : Trigger에 등록된 대상 Trigger의 이름입니다.
- Scheduled : 대상 Trigger가 수행 예정인 시간을 표시합니다.
- Start : 대상 Trigger가 실제로 수행을 시작한 시간을 표시합니다.
- End : 대상 Trigger가 수행 종료한 시작을 표시합니다.
- Elapse : 대상 Trigger가 Start End까지 수행하는데 소모된 시간을 표시합니다.
- **Status** : 대상 Trigger의 현재 상태를 표시합니다.
 - 🥥 Run : 대상 Trigger가 현재 수행 중인 상태입니다.

- ◎ **Complete** : 대상 Trigger가 정상적으로 수행 종료되었다는 상태입니다.
- ◎ 🥝 Aborted :대상 Trigger가 사용자에 의해 강제 종료 되었다는 상태입니다.
- ● Complete :대상 Trigger가 수행 중 오류가 발생하여 비정상적으로 종료되었다는 상태입니다.
- **Result**: Trigger의 수행 결과를 표시하는 컬럼입니다. 대상 Trigger의 Status가 Complete 인 경우에만 기록이 됩니다.
 - SUCCESS : 대상 Trigger 가 정상적으로 수행 종료 되었을 시 SUCCESS로 기록이 됩니다.
 - FAIL : 대상 Trigger가 수행 중 오류가 발생하여 비정상적으로 종료 되었을 시 FAIL로 기록이 됩니다. (사용자가 강제로 종료한 Aborted status는 Result에 기록되지 않습 니다.)
- Message : Trigger가 수행 중 오류가 발생하였을 시 오류 메시지를 기록하는 컬럼입니다.

상단 메뉴 바

 Stop Trigger : Status가 Run 중인 Trigger를 사용자가 강제로 종료하기 위한 버튼입니다. Monitoring의 Stop Trigger 또는 Trigger의 Stop task if it runs longer than 설정에 의해 Trigger가 종료되는 경우 Trigger 종료되면서 Status는
 Aborted 로 상태 변화가 됩니 다.

Stop Trigger, Stop task if it runs longer than (**Triggers 설정 기능**) 기능을 사용하여 Trigger를 중지하는 경우 Thread에 따라서 **@** Complete (FAIL) 상태로도 표시 될 수 있습 니다.

- Auto Refresh Interval : Monitoring Table의 상태를 업데이트 하는 주기를 설정합니다.
 Normal : 주기의 기본 값이며 30초 단위로 Monitoring Table 정보를 업데이트 합니
 - 다.
 - High: 10초 단위로 Monitoring Table의 정보를 업데이트 합니다.
 - Low : 60초 단위로 Monitoring Table의 정보를 업데이트 합니다.
 - Pause : Monitoring Table의 정보 업데이트를 일시적으로 중단합니다.

• Query : Errors 또는 사용자 지정 View에서 사용하는 기능이며, Search Option에 조회하 고자 하는 기간을 입력하고 Query를 실행 시 입력한 기간 만큼의 Monitoring 정보를 표시 합니다.

Performance Trend

Performance Trend에서는 \$model, \$cola Job Type 유형만 집계가 됩니다. Monitor의 Performance Trend 그래프에서는 선택한 대상 Trigger가 수행 시작한 시점의 기준선이 그려 지며 기준선 기준으로 전후 10분 동안의 Server Machine의 CPU/Memory 사용량 및 해당 시 간의 수행 중이었던 Trigger의 개수를 표시합니다. Trigger의 Count는 반드시 사용자가 선택 한 Trigger가 아니더라도 해당 시간에 수행 중이었던 다른 Trigger도 Count도 포함이 됩니다. 즉, 사용자가 조회한 Trigger가 A이고 조회한 시간에 Trigger B가 수행 중이었으면 Performance Trend에서는 해당 시간에 Trigger의 Count는 2가 됩니다.

Performance Trend에서 보이는 CPU/Memory 사용량은 Trigger(Mozart Agent) 외에 다른 프 로세스들이 사용한 CPU/Memory도 포함되고 있습니다. Performance Trend 그래프의 점선 들은 Server Machine의 평균 CPU/Memory 사용량이며 실선은 해당 시간의 사용량을 의미합 니다. 회색 영역형 그래프는 해당 시간에 Trigger 수행 여부 및 개수를 의미합니다. Performance Trend 그래프에 마우스 오버 시 해당 시간의 상세 정보를 조회 할 수 있습니다.

Performance Trend는 1분 주기로 성능 집계를 합니다. 만약 Trigger 수행 시간이 1분 미만에 집계 주기와 중첩되지 않을 시 해당 Trigger 수행 이력 (성능, Count)가 집계되지 않을 수 있습 니다.

Log Files

Log Files에서는 Monitoring에서 선택한 Trigger의 수행 이력 로그 파일들을 조회 할 수 있습니다. Monitoring 탭에서 Trigger 선택 후 상단 메뉴의 [Open Log Folder]를 클릭하면 Log 파일들을 조회 할 수 있는 탭이 활성화 됩니다.

Log Files				ıх
Open Folder				
Name	Date	Size	Attributes	
task-TSK-20170504-134752.log	2017-05-04 13:48:13	1 kb	-9	
persists-TSK-20170504-134752.log	2017-05-04 13:48:12	138 bytes	-a	
task-TSK-20170504-132752.log	2017-05-04 13:28:00	1 kb	-9	-

Trigger Execution Log

Trigger Execution Log에서는 사용자가 선택한 대상 Trigger의 수행 단계 별 소모 시간을 확인 할 수 있습니다. Monitoring에서 Trigger Execution Log에 기록하기 위해서는 Trigger Argument 중performance-profiling이 True이어야 합니다.

- **DOWNLOAD** : Task 수행 중 DB로부터 Data를 Download하는데 소모된 총 시간을 표시 합니다.
- PERSIST_IN : Task 수행 중 Input Data를 로딩하는데 소모된 총 시간을 표시합니다.
- ENGINE_RUN : Task 수행 중 모듈(Pegging, Simulation, CBS, etc)를 수행의 소모 시간을 표시합니다.
- PERSIST_OUT : Task 수행 중 엔진 결과를 저장하는데 소모된 총 시간을 표시합니다.
- SAVE_DB : Task 수행 중 결과 데이터를 대상 DB에 Upload하는데 소모된 총 시간을 표시 합니다.

Error Message

Error Message는 수행 중인 Trigger가 오류가 발생하여 비정상적으로 종료되었을 시 오류에 대한 상세 메시지를 기록합니다.

Error Message를 Clipboard에 복사를 하거나 상세 오류 메시지를 확인하는 작업을 수행 할 수 있습니다.

Error Message:	
System.NullReferenceException: 개체 참조가 개체의 인스턴스로 설정되지 않았습니다. 위치: Fab.Planning.MonitorLogger.Write(String state) 파일 D:\TestProject\Micron.Fab\Test\Fab.Planning\My Methods\Logger\MonitorLogger.cs:줄 23 위치: Fab.Planning.Logic.Main.PROGRESS_REPORT0(ModelContext context, String stage, Boolean& handled) 파일 D:	*
Copy to clipboard Details	

Error 열람 기능

Mozart Management Console (2.0)에서는 엔진 수행 도중 오류가 발생 시 Errors view에 기록 만 할 뿐 아니라 사용자가 오류 발생을 인지 할 수 있도록 알림이 팝업이됩니다.



Error 알림의 경우 MMC2에서 Monitoring의 Errors View 탭이 활성화가 되어 있어야 알림 팝 업이 나옵니다. 또한, RUN 상태에서 오류가 발생 시 팝업이 나오며 엔진이 시작하자마자 종 료되는 경우에는 팝업 메시지가 출력되지 않습니다.

Log 보존 기간 설정

Trigger 가 수행한 이력 정보와 Performance 기록 정보의 보관 기간을 지정 할 수 있습니다. Monitor 우측 마우스를 눌러 Log Options 메뉴를 통해 설정할 수 있습니다.

> 📑 Monitor	
🔀 Perform	Add View
🦉 Shortcu	Log Options
🔄 🖾 SyncPill	

기본 설정된 보관 기간은 60일로 되어 있으며, 보관 기간을 일자(Day) 기준으로 재설정 할 수 있습니다.

Log Options			_		×
- Max Archive Days Se	tting				
Performance:	60				-
Trigger Execution:	60				▲ ▼
Trigger Run Time:	60				•
		(ОК	Cance	:I

Monitoring View 사용법

Monitoring 사용자 View 추가하기

1. Server Explorer에서 View 등록 대상 Server를 선택합니다.

2. 대상 Server의 Monitoring 노드에서 마우스 오른쪽 버튼 클릭하여 **[Add View]** 를 선택합니 다.

3. Add View Dialog에서 View Name에 이름을 입력하고, Monitoring 할 Trigger의 체크박스에 체크합니다. (복수 선택 가능)

	III Add View									
١	View Name: MyView									
	Targ	jet Trig	gers:							
			Name		Descrip	tion				
			Colabo	oration						<u> </u>
	F		Failure	9						
			Fab							
		\checkmark	Deper	ndent						
		\checkmark	Fab3							
			DataB	lackup						
			Test							
			Serve	rTest1						-
								ОК	Canc	el

4. 설정이 완료 되었으면 [OK] 버튼을 클릭합니다.

Monitoring 사용자 View 수정하기

1. Server Explorer에서 View 수정 대상 Server를 선택합니다.

2. 대상 Server의 Monitoring 노드에서 편집하고자 하는 View를 선택하여 마우스 오른쪽 버튼 클릭하여 **[Edit View]** 를 선택합니다. (All,Errors View는 기본 제공 View라 수정이 불가합니 다.)

3. Edit View Dialog에서 내용 수정 후 **[OK]** 버튼을 클릭합니다.

Monitoring View 삭제 방법

- 1. Server Explorer에서 View 삭제 대상 Server를 선택합니다.
- 2. 대상 Server의 Monitoring 노드에서 삭제하고자 하는 View를 선택하여 마우스 오른쪽 버튼
- 클릭하여 **[Delete View]** 를 선택합니다.
- 3. 팝업창에서 [예(Y)] 버튼을 클릭하면 대상 View가 제거됩니다.

Performance 사용법

MMC2의 Performance 노드에서는 대상 Server에 등록된 Trigger들의 전반적인 구동 성능을 모니터링 하는 영역입니다. Trigger Performance를 통해서 성능을 분석하여 엔진 구동 시 Task의 어떤 단계에서 Delay가 발생하는지 파악이 가능합니다. 특히 개발자나 관리자들은 해당 데이터를 통해서 Vmodel 또는 DLL이 변경 전과 후의 성능 비교가 가능하여 개선 여부 를 파악하는데 용이합니다. 다음은 대상 Server에 등록된 Trigger의 Performance를 보는 방법 과 Performance 창의 기능에 대해서 기술합니다.

대상 Trigger Performance 기록 방법

대상 Server에 등록된 Trigger의 성능을 Performance에서 확인하기 위해서는 Performance에 기록하기 위한 Trigger Argument를 설정해야 합니다. Trigger의 Argument는 다음과 같이 설 정이 가능합니다.

1. Server Explorer에서 Trigger Argument를 설정 할대상 Server를 선택합니다.

2. 대상 Server의 Trigger 노드를 더블 클릭하여 Trigger 창을 활성화 합니다.

3. Performance에 기록할 대상 Trigger를 선택하여 더블 클릭 혹은 상단 메뉴 바의 [Edit] 버튼 을 클릭합니다.

4. Edit Trigger Dialog에서 Target Job 탭으로 이동합니다.

5. Target Job에 보이는 리스트 하단으로 내려가서performance-profiling 체크 박스에 체크 후 **[OK]** 버튼을 클릭하여 설정을 저장합니다.

Performance UI 설명

• Job Summary

Job Summary에서는 대상 Server에 등록된 총 Job의 개수와 Job과 매핑된 Trigger의 개 수를 확인 할 수 있습니다.

J	Job Summary						
	Item	Count					
÷	Total Job	19					
	Trigger mapped Job	17					
	Trigger un-mapped Job	2					

- Total Job : 대상 Server에 등록된 Job의 총 개수입니다.
- Trigger mapped Job : 대상 Server에 등록된 Job 중 Trigger가 등록된 Job의 개수입니다. Trigger mapped Job Row를 더블 클릭 시 Trigger가 등록된 Job의 목록을 확인 할 수 있 습니다. 하나의 Job에 Trigger가 여러개 매핑이 되어 있어도 Job의 Trigger 매핑 여부를 집계하기 때문에 Trigger mapped Job에서의 카운트는 1개만 증가합니다. (EX: Job A {Trigger A, Trigger B, Trigger C} -> Trigger mapped Job = 1)

Trigger mapped Job
Job Name
Server Test 2.ServerTest3
Server Test 2.ServerTest4
LoadTest1.LoadTest1
LoadTest2.LoadTest2
LoadTest3.LoadTest3
LoadTest_Monitor.LoadTest_Monitor
LoadTest_Executor.LoadTest_Executor
LoadTest4.LoadTest4
v
H4 44 4 Record 7 of 17 ► H+ H+ - A 🗸 🗙
Close

- Trigger un-mapped Job : 대상 Server에 등록된 Job 중 Trigger가 등록되지 않는 Job의 개수입니다. Trigger un-mapped Job Row를 더블 클릭 시 Trigger가 등록되지 않는 Job 의 목록을 확인 할 수 있습니다.
- Trigger Summary

Trigger Summary에서는 대상 Server에 등록된 Trigger의 요약 정보입니다. 대상 Server 등록된 Trigger 중 활성/비활성화 되어 있는 Trigger의 개수 및 Performance에 기록 대상 Trigger 개수 등의 정보를 확인 할 수 있습니다.

Т	rigger Summary	
	Item	Count
×	Total Trigger	20
	Active Trigger	14
	Profiling Trigger	6

- **Total Trigger** : 대상 Server에 등록된 Trigger의 총 개수입니다.
- Active Trigger : 대상 Server에 등록된 Trigger 중 [Enabled] 가 활성화 된 Trigger의 개수 입니다.
- **Profiling Trigger** : 대상 Server에 등록된 Trigger 중performance-profiling이 True인 Trigger들의 개수입니다.

Trigger Performance

Trigger Performance에서는 Profiling Trigger의 대상 Trigger들의 성능집계 정보를 수치 및 그 래프로 표현하는 영역입니다. Task가 실행이 되었을 시 각 단계별 소모 시간, Trigger의 총 실 행 회수 및 성공/실패 비율 등의 정보를 확인 할 수 있습니다. 다음 은 Trigger Performance 영 역의 용어 정의를 기술합니다.

- Period : 사용자가 Trigger Performance를 조회하고자 하는 기간을 설정합니다. 기간은 Days/Hours 중 선택이 가능하며 현재를 기준으로 설정한 Days/Hours 전에 집계된 이력 들을 호출합니다.
- Trigger
 - Name : Triggers 노드 기준 등록된 Trigger의 명칭입니다.
 - Description : 대상 서버의 등록된 Trigger의 설명입니다. Description은 Triggers 노드 에서 입력이 가능합니다.

Trigger	
Name 🔺	Description
Colaboration	
Dependent	Dependent Trigger Test
Fab	
Fab3	
FabCopyTest	

• RunTime(sec)

- DOWNLOAD : Task 수행된 총 횟수의 DB로부터 Data를 Download하는데 소모된 평 균 시간을 표시합니다.
- PERSIST_IN : Task 수행된 총 횟수의 Input Data를 로딩하는데 소모된 평균 시간을 표시합니다.
- ENGINE_RUN : Task 수행된 총 횟수의 모듈(Pegging, Simulation, CBS, etc)를 수행
 의 평균 소모 시간을 표시합니다.
- PERSIST_OUT : Task 수행된 총 횟수의 엔진 결과를 저장하는데 소모된 평균 시간을 표시합니다.
- SAVE_DB : Task 수행된 총 횟수의 결과 데이터를 대상 DB에 Upload하는데 소모된 평균 시간을 표시합니다.
- TOTAL_RUN : DOWNLOAD SAVE_DB Action 까지의 누적 시간의 평균 시간입니다.

Run Time(sec)								
DOWNLOAD	PERSIST_IN	ENGINE_RUN	PERSIST_OUT	SAVE_DB	TOTAL_RUN			
00:00:00	00:01:18	00:04:04	00:00:02	00:00:00	00:06:07			
00:00:00	00:03:26	00:00:21	00:00:02	00:00:00	00:04:09			
00:00:00	00:00:49	00:02:25	00:00:02	00:00:00	00:03:37			
00:00:00	00:00:41	00:00:20	00:00:00	00:00:00	00:01:04			
00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:02			

Trigger Performance의 RunTime(sec)에서는 대상 Trigger의 Action 별 평균 소모 시간을 표시가 되며 Trigger Performance 영역의 좌측 하단 그리드에서는 대상 Trigger의 Action 별 최대/최소 소모시간 조회가 가능합니다.

Action	Min	Average	Max
PERSIST_IN	00:00:49	00:01:18	00:01:32
ENGINE_RUN	00:02:29	00:04:04	00:04:32
PERSIST_OUT	00:00:01	00:00:02	00:00:03

또한, Trigger Performance 영역의 우측 하단에서는 사용자가 설정한 기간에 대상 Trigger의 실행 횟수만큼 각 Action 별 TOTAL_RUN 시간에서 차지하는 비율을 그래프 형태로 확인이 가 능합니다. 상세한 정보 조회를 위해서는 조회하고자 하는 바 그래프에 마우스 오버 시 해당 시간에 수행된 Trigger의 Action별 수행 시간을 확인 할 수 있습니다.



- Reliability
 - MIN: 대상 Trigger 총 실행 횟수 중 Task 수행 시간이 가장 짧은 시간입니다.
 - MAX : 대상 Trigger 총 실행 횟수 중 Task 수행 시간이 가장 긴 시간입니다.
 - LIMIT : 사용자가 설정한 대상 Trigger의 수행 제한 시간입니다. Triggers -> 대상 Trigger -> Schedule Tab -> Stop task if it runs longer than 옵션에 설정된 시간입니 다.
 - RUN_COUNT : 대상 Trigger가 실행된 총 횟수입니다. Stop task if it runs longer than 옵션 또는 Monitoring에서 Stop Trigger 기능을 통해 사용자가 강제로 중지한 경우 집계되지 않습니다.
 - FAIL_COUNT : 대상 Trigger가 총 실행된 횟수 중 오류에 의해 비정상적으로 중지된 횟수입니다.
 - FAIL_RATE : 대상 Trigger의 RUN_COUNT 중 FAIL_COUNT의 비율입니다.
 - SUCCESS_RATE : 대상 Trigger의 RUN_COUNT 중 정상 수행된 비율입니다.

Reliability								
MIN	MAX	LIMIT	RUN_COUNT	FAIL_COUNT	FAIL_RATE	SUCCESS_RATE		
00:03:35	00:07:06	00:00:00	5	0	0 %	100 %		
00:04:09	00:04:09	00:00:00	1	0	0 %	100 %		
00:00:01	00:07:37	00:02:00	22	6	27 %	73 %		
00:01:04	00:01:04	00:00:00	1	1	100 %	0 %		
00:00:02	00:00:02	00:00:00	1	1	100 %	0 %		

• **Trigger Performance Dialog (Trigger의 대상 Server Machine의 자원 사용량 확인하기)** Performance 창에서는 Trigger가 수행된 시간 동안 대상 Server Machine의 CPU/Memory 사용량 및 비율을 그래프 형태로 Trigger Performance Dialog에서 조회가 가능합니다. *Performance 창의 Trigger Performance 그리드에서 대상 Trigger의 Row 를 더블 클릭 시* Trigger Performance Dialog 창이 열립니다.



Period는 Trigger Performance에서 사용자가 지정한 Period 기준을 따르고 있으며, Trigger Performance Dialog 창에서 사용자가 임의로 기간을 다시 설정 할 수 있습니다. Trigger Performance Dialog 창에서는 대상 Trigger가 수행된 시점에 수행 시간 만큼 집계가 됩니다. Trigger 수행 시간 동안의 Server Machine의 총 CPU/Memory 사용량을 그래프로 표시되며 상단의 푸른색 그래프는 CPU 사용량, 하단의 녹색 그래프는 Memory 사용량입니다. 그 중 연 한 푸른색은 CPU 총 사용량 중 대상 Trigger (MozartAgent 프로세스)가 차지하는 비율이고, 연한 녹색은 Memory 총 사용량 중 대상 Trigger (MozartAgent 프로세스)가 차지하는 비율이고, 인한 녹색은 Memory 총 사용량 중 대상 Trigger (MozartAgent 프로세스)가 차지하는 비율입 니다. 진한 색상의 경우는 Trigger가 차지하는 비율 외에 다른 프로세스들이 차지하는 비율을 나타냅니다. MMC2에서의 대부분의 그래프와 마찬가지로 Trigger Performance Dialog 역시 그래프 바 위에 마우스 오버 시 상세한 정보 확인이 가능합니다.

Log 보존 기간 설정

Trigger 가 수행한 이력 정보와 Performance 기록 정보의 보관 기간을 지정 할 수 있습니다. Performance 우측 마우스를 눌러 Log Options 메뉴를 통해 설정할 수 있습니다.

🔀 Performance	
🔄 Shortcuts	Open
🧑 SyncProjects	Log Options
🥵 lisers 🛛 🖳	

기본 설정된 보관 기간은 60일로 되어 있으며, 보관 기간을 일자(Day) 기준으로 재설정 할 수 있습니다.

E Log Options			_		×
– Max Archive Days Se	tting				
Performance:	60				•
Trigger Execution:	60				•
Trigger Run Time:	60				•
		0	ĸ	Cance	2

Backup 사용법

Server에서 여러 Job/Trigger를 등록하여 Task를 수행하게 될 시 엔진 결과, 로그 파일 등에 의해 WorkingDirectory가 위치한 HDD 용량이 가득 찰 수 있습니다. 이는 관리자가 별도로 신 경을 써서 관리하지 않는 이상 HDD 용량 부족 현상으로 엔진 결과 등이 저장 되지 않을 수 있 습니다. MMC2에서는 사용자가 효율적인 HDD 용량 관리를 위해 Backup을 Scheduling 할 수 있도록 기능을 제공합니다. MMC2 Backup 기능을 통해 다음과 같이 설정이 가능합니다.

- Backup 주기 : Task Trigger를 설정하는 것과 같이 Backup도 사용자가 Backup 작업을 수 행 할 수 있는 주기를 설정 할 수 있습니다.
- Backup 경로 : WorkingDirectory가 위치한 다른 드라이브의 HDD로 경로 설정이 가능합니다.
- Source 파일 유지 여부 : Backup 작업을 진행 후 Source 파일을 유지/삭제 여부를 결정 할 수 있습니다.
- 파일 필터링 : Source 경로에서 특정 확장자만 백업 진행/예외 여부를 설정 할 수 있습니다.
- 파일 크기 필터링 : Backup을 진행 할 파일 최대/최소 크기를 설정 할 수 있습니다.
- 특정 기간의 파일/폴더 Backup : MMC2 Backup 기능을 통해 사용자가 설정한 특정 기간 이후의 파일/폴더들 Backup하도록 설정 가능합니다.

MMC2에서 Backup 설정에 대한 상세한 내용은 Backup 등록/수정 방법에서 확인이 가능합니다.

Backup 등록/수정 방법

- 1. Server Explorer에서 Backup 등록 대상 Server를 선택합니다.
- 2. 대상 Server의 Backup을 더블 클릭하여 Backup 조회화면을 활성화 합니다.
- 3. 상단 메뉴바에서 [Add] 버튼을 클릭합니다.
- 4. New Backup Dialog에서 Schedule, Settings Tab 별 정보를 입력합니다.
- 5. Schedule Tab 정보부터 입력합니다. 기본적인 설정은 Trigger 등록 방법과 동일합니다.

🔡 New Backup Sched	lle		
Schedule Setting:			
Trigger name: Category: Description:	TestBackup		
Settings One Time Simple Daily Weekly Monthly Dependent	Start: 2017-04-13 (Expire: 2017-04-13 (■▼ 오후 6:21:19 🚔 Set Now + 5: ■▼ 오후 6:21:19 🚔	380
Advanced Settings			
 Priority: Stop task if it ru Enabled Trace 	ns longer than:	0 Retry Interval: 00:00:00 Retry Count:	00:00:00
Schedule Now			OK Cancel

6. Settings Tab 정보를 입력합니다.

🔡 New Backup Schedule	
Schedule Settings	
- Folder Pairs	
	+ -
Policy	
Remove source files after backup	
Backup files as ZIP compression	
Include hidden files and folders	
Filter	
File size	
Min size:	
File Type filter:	
🗖 Last Write Time: within 1 🚔 Hours 👻	
Log	
Save Log	
Log Directory:	
OK OK	Cancel

- Folder Pairs : Backup의 Source 폴더 및 Backup 할 Destination 경로를 지정합니다. → 클릭 시 Source/Destination 경로를 추가를 하고 → 로 Folder Pairs에 등록된 Source/Destination을 삭제 할 수 있습니다.
- Policy : Backup 정책을 설정합니다. 복수 선택이 가능합니다.
 - Remove source files after backup : Backup 후 Source 경로에 있는 파일/폴더 삭제 여부를 체크합니다.
 - Backup files as ZIP compression : Backup Source를 zip 파일로 압축하여 Destination에 Backup 여부를 체크합니다.
 - Include hidden files and folders : Source 경로의 숨 파일도 Backup을 할지 여부를 체크합니다.
- Filter : Backup의 Filter 조건을 설정합니다. 복수 설정이 가능합니다.
 - File size : 체크 시 Backup의 최대/최소 파일 크기를 설정 할 수 있습니다.
 - Min size : 입력된 크기 보다 작은 파일들을 백업하지 않기 위해 설정합니다. (단 위 : KB/MB/GB)

- Max size : 입력된 크기 보다 큰 파일들을 백업하지 않기 위해 설정합니다. (단위 : KB/MB/GB)
- File Type filter : 체크시 해당 확장자의 파일만 Backup 또는 제외할지 설정합니다.
 - including: including을 선택하고 text box에 파일 확장자 입력 시 Source 경로에 있는 해당 확장자의 파일만 Backup을 진행합니다. 복수 개의 파일 입력 시 ','로 구분자를 사용합니다.
 - excluding : excluding을 선택하고 text box에 파일 확장자 입력 시 Source 경로 에 있는 해당 확장자의 파일을 제외한 나머지 파일들을 Backup합니다. 복수 개 의 파일 입력 시 \? 구분자를 사용합니다.
- Last Write Time : 특정 기간 이상이 된 파일들만 Backup을 진행하기 위해 설정합니다. 단위는 시간/일입니다. (i.e 1 Hour로 설정 시 현재 시간으로부터 파일의 최종 수정 날짜가 1시간 이상 된 파일들만 백업을 합니다.
- Log : Backup 수행에 대한 Log를 기록할지에 대한 여부를 설정합니다.
 - Save Log : 체크 박스가 활성화 시 Backup에 대한 로그를 기록합니다.
 - Log Directory : Backup 수행의 Log 파일 저장 경로를 설정합니다. Backup Log 파일
 은 .log 포맷의 텍스트 문서로 저장이 되며 파일 이름 형식은 backup-[Backup
 Name]-yyyymmdd-hhmmss.log입니다.

🔇 - 🕥 - 📄 🕞 workingDirectory#Logs#E	BackupLog₩	😂 🚺 🚹	
Name	Date	Size	Attributes
backup-DataBackup-20170328-233917.log	2017-03-28 23:39:18	125 bytes	-9
backup-DataBackup-20170328-233941.log	2017-03-28 23:39:43	1 kb	-a
backup-DataBackup-20170328-234027.log	2017-03-28 23:40:29	196 bytes	-a
backup-DataBackup-20170328-234102.log	2017-03-28 23:41:04	125 bytes	-a
backup-DataBackup-20170328-234127.log	2017-03-28 23:41:29	1 kb	-a
backup-DataBackup-20170328-234149.log	2017-03-28 23:41:50	336 bytes	-a
backup-DataBackup-20170328-234336.log	2017-03-28 23:43:37	336 bytes	-a
backup-DataBackup-20170328-234425.log	2017-03-28 23:44:27	1 kb	-a
backup-DataBackup-20170328-234444.log	2017-03-28 23:44:46	1 kb	-a
backup-DataBackup-20170328-234511.log	2017-03-28 23:45:12	1 kb	-a
backup-DataBackup-20170328-234532.log	2017-03-28 23:45:34	196 bytes	-a
backup-DataBackup-20170328-234648.log	2017-03-28 23:46:50	530 bytes	-9
backup-DataBackup-20170328-234811.log	2017-03-28 23:48:12	265 bytes	-a
backup-No Name-20170328-233632.log	2017-03-28 23:36:34	119 bytes	-9

▲ Backup 수행 이력 Log 저장경로는 WorkingDirectory가 아니어도 상관이 없습니다. 이에 Projects 처럼 Log를 확인 할 수 있는 노드가 Server Explorer에 추가되지 않으며 사용자가 Shortcut으로 등록 아니면 Backup Log를 저장한 경로에 직접 접근하여 이력을 조회 할 수 있 습니다.

Backup 확인하기

- 1. Server Explorer에서 Backup 확인 대상 Server를 선택합니다.
- 2. 대상 Server의 Backup을 더블 클릭하여 Backup 조회화면을 활성화 합니다.
- 3. Backup 리스트 중 Backup 파일/폴더를 조회하고 싶은 Backup Trigger를 클릭합니다.
- 4. 하단 Destination Folders Tab에서 선택된 Backup의 파일 및 폴더를 확인 할 수 있습니다.

Backup 수정하기

- 1. Server Explorer에서 Backup 수정 대상 Server를 선택합니다.
- 2. 대상 Server의 Backup을 더블 클릭하여 Backup 조회화면을 활성화 합니다.
- 3. Backup 리스트에서 수정 대상 Backup Trigger를 더블 클릭합니다.
- 4. Schedule/Settings Tab 정보를 수정하여 **[OK]** 버튼을 클릭합니다.

Backup 삭제하기

- 1. Server Explorer에서 Backup 삭제 대상 Server를 선택합니다.
- 2. 대상 Server의 Backup을 더블 클릭하여 Backup 조회화면을 활성화 합니다.
- 3. Backup 리스트에서 삭제를 진행할 Backup Trigger를 선택합니다.
- 4. 상단 메뉴바의 **[Remove]** 버튼을 클릭하여 Backup을 삭제합니다.

사용자 등록 및 권한 설정

사용자 계정 및 권한

MMC에서는 사용자의 역할 별 권한을 부여하여 대상 Server에서 수행 할 수 있는 작업을 제 한하고 있습니다. 모든 사용자가 동일한 권한을 가지고 동일한 기능을 사용하게 될 시 인위적 인 실수의 발생 가능성이 높아지고 원인 추적이 어려울 수 있습니다. 이러한 어려움을 해소하 기 위해 MMC에서는 사용자 역할에 따라 관리자 (Administrator), 개발자 (Developer), 운영자 (Operator), 관람자 (Viewer) 권한을 분류하여 권한에 따라 MMC 기능 사용을 제한하고 있습 니다. MOZART Server 설치 시 기본적으로 관리자 권한 계정이 생성이 됩니다. (서버 관리 참 고)

- Administrator : MMC의 모든 기능을 사용 할 수 있는 권한입니다. 모든 권한 중 유일하게 사용자 계정 생성/편집 및 권한 부여를 할 수 있습니다.
- **Developer** : 개발자 권한이며, 해당 권한부터 Project를 관리 할 수 있으며 Project 생성/ 편집/파일 배포 등의 작업을 수행 할 수 있습니다.
- **Operator** : 주로 Job/Trigger를 수행/관리하기 위한 권한입니다. 해당 권한에서는 Project 생성/편집/파일 배포를 할 수 없습니다.
- Viewer : 생성/편집/삭제에 대한 어떠한 작업도 수행 하지 못하고 관람만 가능한 권한입니다.

각 권한 별 MMC에서 수행 할 수 있는 작업에 대한 상세한 내용은 아래와 같습니다.

기능	Viewer	Operator	Developer	Administrator
서버 접속	0	0	0	0
서버 접속 정보 수정	0	0	0	0
서버 접속 해제	0	0	0	0
프로젝트 목록 보기	0	0	0	0
Job 노드 표시	0	0	0	0
Trigger 노드 표시	0	0	0	0
System Log 노드 표시	0	0	0	0
Backup 노드 표시		0	0	0
Monitor 노드 표시	0	0	0	0
Performance 노드 표시	0	0	0	0
Shortcut 노드 표시	0	0	0	0
User 노드 표시		0	0	0
Deployable Folder 노드 표시	0	0	0	0

JobScheduler 상태 보기	0	0	0	0
JobScheduler 중지		0	0	0
JobScheduler 재개		0	0	0
서버 라이선스 보기	0	0	0	0
서버 스펙 보기	0	0	0	0
프로젝트 추가			0	0
프로젝트 보기	0	0	0	0
프로젝트 편집			0	0
프로젝트 제거			0	0
프로젝트 배포			0	0
프로젝트 변경 이력 보기	0	0	0	0
Job 목록 보기	0	0	0	0
Job 보기	0	0	0	0
Job 추가/편집/제거		0	0	0

Job 변경 이력 보기	0	0	0	0
Job Type 목록 보기		0	0	0
Job Type 추가/편집/제거		0	0	0
Trigger 목록 보기	0	0	0	0
Trigger 보기	0	0	0	0
Trigger 추가/편집/제거		0	0	0
Trigger 변경 이력 보기	0	0	0	0
Trigger 실행 이력 보기	0	0	0	0
Backup Trigger 목록 보기		0	0	0
Backup Trigger 보기		0	0	0
Backup Trigger 추가/편집/제거		0	0	0
Destination Folder 보기		0	0	0
Destination Folder 파일 다운로드		0	0	0
Trigger View 추가/편집/제거	0	0	0	0

Trigger View 보기	0	0	0	0
Trigger 강제 중지		0	0	0
Trigger Performance 보기	0	0	0	0
Shortcut 보기	0	0	0	0
Shortcut 추가/편집/제거	0	0	0	0
Shortcut 파일 다운로드	0	0	0	0
사용자 목록 보기		0	0	0
사용자 추가/편집/제거				0
사용자 추가/편집/제거 사용자의 변경 이력 보기		0	0	0
사용자 추가/편집/제거 사용자의 변경 이력 보기 Deployable Folder 보기	0	0	0	0 0 0
사용자 추가/편집/제거 사용자의 변경 이력 보기 Deployable Folder 보기 Deployable Folder 파일 다운로드	0	0 0 0	0 0 0	0 0 0 0
사용자 추가/편집/제거 사용자의 변경 이력 보기 Deployable Folder 보기 Deployable Folder 파일 다운로드 Deployable Folder 파일 업로드	0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
사용자 추가/편집/제거 사용자의 변경 이력 보기 Deployable Folder 보기 Deployable Folder 파일 다운로드 Deployable Folder 파일 업로드 Deployable Folder 파일 제거	0 0 0	0 0 0 0	0 0 0 0	0 0 0 0 0

사용자 계정 생성 및 권한 설정에 대한 자세한 설명은 <mark>사용자 등록/수정 방법</mark>에서 참고 바랍 니다.

사용자 등록/수정 방법

사용자 등록하기

1. Server Explorer에서 사용자를 등록할 대상 Server를 선택합니다.

2. 사용자 등록을 위해서는 대상 Server에 접속 시 반드시 관리자 (Administrator) 계정으로 접 속합니다.

3. 대상 Server의 Users를 더블 클릭하여 Users 화면을 활성화합니다.

4. 상단 메뉴바에서 [Add] 버튼을 클릭합니다.

5. Add new user Dialog에서 필요한 정보를 입력합니다.
| 🗱 Add new user | | x |
|-----------------------|-----------------------|---|
| Input user informatio | n. | _ |
| User ID : | Admin | * |
| Password : | ***** | * |
| Password again : | ***** | * |
| First Name : | Andrew | |
| Last Name : | Park | |
| Email : | andpark@mozart.com | |
| Role : | Administrator 💌 | * |
| | Administrator account | |
| Description: | | |
| b computerin | | |
| | | |
| | OK Cancel | |

- User ID : 필수 입력 정보입니다. 대상 서버에 접속 시 사용할 계정을 설정합니다.
 - **Password** : 필수 입력 정보입니다. 대상 서버에 접속 시 사용할 비밀번호를 설정합니다.
 - Password again : 필수 입력 정보입니다. Password text box에 입력된 값과의 일치
 여부를 확인하게 위해 동일한 값을 입력합니다.
 - First Name : 계정 사용자의 이름을 입력합니다.
 - Last Name : 계정 사용자의 성을 입력합니다.
 - Email : 계정 사용자의 이메일 주소를 입합니다.
 - Role : 필수 선택 사항입니다. 해당 계정의 권한을 설정합니다. 권한 별 상세한 내용
 은 사용자 계정 및 권한을 참고 바랍니다.
 - **Description** : 해당 계정에 대한 설명을 입력합니다.
- 6. 정보를 모두 입력하였으면 **[OK]** 버튼을 클릭합니다.

사용자 수정하기

1. Server Explorer에서 사용자를 등록할 대상 Server를 선택합니다.

2. 사용자 등록을 위해서는 대상 Server에 접속 시 반드시 관리자 (Administrator) 계정으로 접 속합니다.

3. 대상 Server의 Users를 더블 클릭하여 Users 화면을 활성화합니다.

4. 리스트에서 편집할 계정을 선택하여 더블 클릭 또는 상단 메뉴바에서 [Edit] 을 클릭합니다.

• 사용자 편집 시 UserID는 편집이 불가합니다. 다만 MOZART Server 설치 시 기본 생성된 계정 'sa' 의 경우는 Role 편집도 불가합니다.

• Password의 경우 입력된 값이 없는 경우 기존에 설정한 Password를 유지합니다.

사용자 삭제

1. Server Explorer에서 사용자를 등록할 대상 Server를 선택합니다.

2. 사용자 등록을 위해서는 대상 Server에 접속 시 반드시 관리자 (Administrator) 계정으로 접 속합니다.

3. 대상 Server의 Users를 더블 클릭하여 Users 화면을 활성화합니다.

4. 리스트에서 삭제를 할 계정을 선택하고 [Remove] 버튼을 클릭합니다.

▲ MOZART Server 설치 시 기본 생성된 'sa' 계정은 삭제 할 수 없습니다.

MOZART 서버 기본 관리자 계정 암호 재설정 방법

기능 설명

Mozart Server 설치 시 기본으로 제공되는 관리자 계정 ("sa")의 암호를 잊었을 시 Server Configurator를 통해 재설정 할 수 있습니다.

B Mozart Configurator for Server		_		×
Product Management	Management Service Configuration			
License				
Service Monitoring	Service Port			
Packages	8000			
Configuration	JobScheduler Channel Port			
MozartJobScheduler	8886			
MozartManagementSer				
MozartServerService	Working Directory			
	D: ₩vms₩mozart		Brow	vse
	Default Shortcuts			
	Name Directory Editable		Ad	d
			Ed	it
			Rem	ove
			Rent	0.00
Change password Show Tutorial	Reset "sa" password sa 계정 재설정 버튼			
				a.

기본 관리자 계정의 암호 재설정을 위해 사전에 알아두어야 할 사항은 다음과 같습니다.

- 암호 재설정 기능을 사용하기 위해서는 최소한 MozartManagementService 패키지는 설 치되어 있어야 합니다.
- 기본 관리자 계정인 "sa"의 암호만 재설정 할 수 있습니다. 그외 사용자에 대한 암호 재설 정은 MMC의 Users를 통해 할 수 있습니다.

• 기본 관리자 계정 암호 재설정은 Mozart Server의 서비스 구동 여부와 관계없이 변경이 가능합니다.

사용 방법

- 1. MMC의 암를 변경하고자 하는 서버 장비에 접속합니다.
- 2. 시작 메뉴에서 Server Configurator 또는 파일 탐색기에서 Mozart Server가 설치된 경로 의 Installer 폴더에서(i.e *C:\Program Files (x86)\VMS\Mozart\Server\Installer*) MozartProductConfigurator.exe 파일을 실행합니다.
- 3. Server Configurator를 시작하기 위한 관리자 암호를 입력하고 [OK] 버튼을 클릭합니다. 기본 암호는 "mozart"입니다.
- 4. Server Configurator가 시작되면 좌측 메뉴에서 [Configuration] > [MozartManagementService]로 이동합니다.
- 5. Management Service Configuration 화면 최하단에 [Reset "sa" password]를 클릭합니다.
- 6. 입력창에 새로 변경할 암호를 입력하고 **[OK]** 버튼을 클릭합니다.
- 7. 마지막 확인 메시지에서 [예(Y)] 버튼을 클릭하여 변경 사항을 저장합니다.

Shortcut 관리

Shortcut 은 MMC 사용자가 Server 폴더에 접근하는 방식 (Shortcut 개념)으로 Server Explorer 를 통해 관리할 수 있습니다.

Shortcut 추가하기

1. Shortcut 을 추가하기 위해 접근대상 Server 노드를 선택합니다.

2. 대상 Server 에서 오른쪽 마우스 버튼을 클릭하여 **[Add Shortcut]** 을 클릭하거나, Server Explorer 상단의 🔃 버튼을 통해 Shortcut 을 추가합니다.

3. New Shortcut 다이얼로그에서 각 항목을 입력합니다.

🖳 New Shortcut	×
Shortcut Name:	ModelFiles
Shortcut Directory:	
Description:	
	OK Cancel

- **Shortcut Name** : Shortcut ID 입니다. 표시되는 Shortcut 명칭입니다.
- Shortcut Directory : Shortcut 과 Mapping 되는 Server 의 폴더입니다. 오른쪽의 [...] 버튼 을 통해 Server 폴더를 선택합니다. Working Directory는 MOZART Server 설치 시 입력한 폴더 입니다. Shortcut 은 Working Directory 하위에서만 설정할 수 있습니다.

💀 Browse For Folder	
WorkingDirectory	
Folder: D:\MozartServer.	
New Folder Delete Folder	OK Cancel

• **Description** : Shortcut 에 대한 설명을 입력합니다.

4. 생성된 Shortcut 은 팝업메뉴의 **[Open]** 메뉴를 실행하거나 더블 클릭하여 접근할 수 있습니다. 사용형식은 Windows 탐색기와 유사하며, 해당 Shortcut 하위폴더로만 탐색이 가능합니다.

🚿 Mozart Management Console				- • ×
File Window Help				
Server Explorer 🛛 🗛 🗙	Jobs Triggers	ModelFiles $ imes$		-
	🕞 • 💮 • 🏂 陵	D:₩MozartServer₩Models₩	4	> ♣ ♣
My Computer (CHUNGG Models Good Triggers Models Models Models Models Models Models Models Models Models	Name SimpleMfg TestModel	Date 2015-01-19 11:14:56 2015-01-19 13:42:22	Size	Attributes
	•			•
Output				
Output : General	•			

Shortcut 수정하기

1. 수정 대상 Shortcut 을 선택하고 오른쪽 마우스 버튼을 이용하여 **[Edit Shortcut]** 팝업메뉴 를 실행합니다.

2. 상기 Shortcut 추가할 때 입력항목을 수정합니다. Shortcut Name 및 Shortcut Directory 등 을 수정합니다.

Shortcut 삭제하기

1. 삭제 대상 Shortcut을 선택하고 오른쪽 마우스 버튼을 이용하여 [Delete Shortcut] 팝업메 뉴를 실행하거나 Server Explorer 상단의 🔃 버튼을 클릭하여 삭제합니다.

Server 기본 Shortcut 설정하기

Shortcut 은 사용자별로 설정하여 사용하게 됨으로 사용하는 PC 별로 설정정보가 저장됩니다. 그러나 Server 에서 공용으로 모든 MMC User 에게 공유할 Shortcut 을 설정하는 경우, Server 에서 Shortcut 공유 여부를 설정 할 수 있습니다.

1. Text editor 로 "mozart.dir" 명으로 빈 파일을 생성합니다.

2. 서버에서 Working Folder 및 그 외 폴더 중 MMC 사용자가 접근 가능하도록 열어줄 폴더를 한 행씩 입력합니다. 입력한 폴더 뒤에 ";true"를 입력하면 해당 경로는 MMC를 통해서 폴더 생성/파일 업로드가 가능합니다. 경로 뒤에 ";true" Argument가 추가되지 않으면 MMC를 통 해 파일 다운로드만 가능합니다. *(e.g D:\modelbackup;true)*



3. 위와 같이 만든 파일을 MOZART Server 가 설치된 폴더*(e.g. C:\Program Files* (x86)\VMS\Mozart\Server or C:\Program Files\VMS\Mozart\Server)에 복사합니다.

4. 그리고 MMC 를 통해 Server 에 접근하면 아래와 같이 사전 정의된 폴더가 별도의 Shortcut 형태로 미리 생성되어 나타납니다. 해당 Shortcut 은 사용자가 추가한 Shortcut 과 동일하게 작동합니다(true가 아닌 경우). Shortcut 에 대한 자세한 내용은 Shortcut 관리를 참고하시기 바랍니다.



사용자 정의 Shortcut 수정하기

1. Text editor로 기존 생성된 "mozart.dir" 파일을 엽니다.

2. 수정하고자 하는 폴더 행을 수정 후 저장합니다.

사용자 정의 Shortcut 삭제하기

1. Text editor로 가진 생성된 "mozart.dir" 파일을 엽니다.

2. 삭제하고자 하는 폴더 행을 삭제 후 저장합니다.

사용자 정의 Shortcut 폴더 생성 및 파일 업로드/다운로드하기

MMC2의 기본 Shortcut에서는 파일 다운로드 외에 할 수 없습니다. "mozart.dir"를 통해 생성 된 경로 또한 같은 규칙을 따르지만 경로 뒤에 ";true" Argument가 추가가되면 폴더 생성 및 파일 업로드/다운로드가 가능합니다.

1. **신규 폴더 생성** : 상단 메뉴의 **[New Folder]** 버튼을 클릭하여 폴더명을 입력하면 폴더를 생 성할 수 있습니다.

192.168.1.10/ManualZip 192.168.1.10/Aut	oUpdate ×	
🔇 - 🕟 - 📄 🔀 D:\AutoUpdate\		🗢 🖬 🖬
Name	Date	Size
TestModel	2017-05-17 16:0	9:47
Create New Folder		
Enter New Folder Name:		
TestModel2	Cancel	

2. **파일 업로드** :

• 상단 메뉴의 [Upload (위쪽방향 화살표)] 버튼을 통해 Upload 다이얼로그를 실행시킵니다.

 +
 [Select Files], [Select Folders], [Clear List] 버튼을 통해 파일을 하단의 리스트에 추

 가/삭제 할 수 있습니다.

Upload Manag	ger		-		
Upload Path	D:\AutoUpdate\				
Select Files	Select Folders Clear Lis	it		(Upload
Name		Last Access Time	Size	File Type	Status
				(<u>C</u> lose

3. 파일 다운로드 :

- Shortcut 메뉴를 통해 Download 대상 폴더를 선택합니다.
- 다운받을 파일을 선택합니다. Windows 탐색기와 같이 마우스로 여러 개의 파일을 Drag 하여 선택할 수 있습니다.
- 상단 메뉴의 **[Download (아래방향 화살표)]** 버튼을 이용하여 다운받을 파일을 저장할 폴 더를 선택합니다.

물더 찾아보기 ····································
🔜 바탕 화면
▷ 🧊 라이브러리
▷ 🔀 VMS
▷ 🖳 컴퓨터
▷ 📬 네트워크
▷ 💷 제어판
· 휴지통
⊳ 퉲 Test
새 폴더 만들기(<u>A</u>) 확인 취소

• 폴더 선택 후 다운로드 받습니다.

라이선스 만료 확인

MOZART Server의 License 만료일이 30일 이내인 경우 MMC에서 License 만료 일정을 팝업 으로 알려줍니다. License를 체크하는 시점은 사용자가 MMC에서 서버 접속 시 또는 Server Explorer에서 License Information을 조회 시 팝업 메시지가 출력됩니다.



서버 접속 시 확인 방법

- 1. Mozart Management Console을 실행합니다.
- 2. Server Explorer에서 대상 Server에 ID/Password 입력 후 접속 합니다.

License Information에서 확인 방법

Server Explorer에서 대상 Server를 선택 후 마우스 우측 버튼 메뉴에서 [License Information] 을 클릭합니다.

i Note:

License 만료 날짜가 지날 시 Job Scheduler, Server Service의 시작 및 대상 서버 접속은 가능하지만 Trigger 수행은 불가합니다. 다만, License가 만료되자마자 Trigger 수행이 중단되는 것이 아니며 만료일로부터 +7일까지는 Trigger 수행이 가능합니다. 만료일로부터 7일 이후에는 Trigger 수행이 중단이 되오니 그 전에 License 갱신을 해야합니다. 만료일이 된 이후에 License 만료일정 조회 시 Trigger 수행 유효 기간이 표시됩니다.

