



serena.com

SERENA® **DIMENSIONS® CM 10.1.1**

Dimensions Build User's and Administrator's Guide

Serena Proprietary and Confidential Information



Copyright © 2001–2007 Serena Software, Inc. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Serena. Any reproduction of such software product user documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

This document contains proprietary and confidential information, and no reproduction or dissemination of any information contained herein is allowed without the express permission of Serena Software.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Trademarks

Serena, TeamTrack, StarTool, PVCS, Collage, Comparex, Dimensions, RTM, Change Governance, and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo, Professional, Version Manager, Builder, Meritage, Command Center, Composer, Reviewer, Mariner, and Mover are trademarks of Serena Software, Inc.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 2755 Campus Drive, San Mateo, CA 94403.

Part number: MA-CMBLD-002

Publication date: April 2007

Table of Contents

Welcome to Serena Dimensions	13
Product Name Changes and New Terminology	14
Printing Manuals	16
Contacting Technical Support	16
Part 1	17
<i>Chapter 1</i>	
C Quick Start, Part One	19
Before You Begin	20
Exercise 1 Create a New Work Area	20
Exercise 2 Display the Dimensions Build Administration Console	22
Exercise 3 Create a Build Configuration	23
Exercise 4 Attach a Build Area	26
Exercise 5 Create the Source File	27
Exercise 6 Add a Build Target	28
Exercise 7 Check in the Build Configuration	35
Exercise 8 Run the Build	36
Exercise 9 View the Execution Details of the Build	40
Launch Log	40
Build Job Details	40
Exercise 10 Re-defining Build Options	42
Exercise 11 Create A Build Configuration Using Copy	44
Copy the Existing Build Configuration	44
Edit the New Build Configuration	44
Check in the New Build Configuration	46
Run the New Build Configuration	46
Summary	46
What's Next	47
Preview of Other Features	47
Additional Information	47
More About Build Areas	47
<i>Chapter 2</i>	
C Quick Start, Part Two	49
Objectives	50
Before You Begin	50
Exercise 1 Create a New Work Area	50
Exercise 2 Add the Payroll Project Build Configuration	52
Exercise 3 Attach the Build Area	53
Exercise 4 Define the Build Target	54
Exercise 5 Define the Build Script	55
Exercise 6 Add Files from Dimensions	56
Exercise 7 Add Existing Sources as Inputs	58
Exercise 8 Check in the Build Configuration	59
Exercise 9 Run the Build	60
Exercise 10 Verify that Target Was Created	62
Exercise 11 Verify that Target Was Uploaded	63

Summary	63
Preview of Other Features	64
Chapter 3 Assembler MVS Quick Start	65
What You Will Learn	65
Before You Begin	66
Pre-Requisites	66
Data Set Naming Convention	66
Member Naming Conventions	67
About Build Templates	67
Exercise 1 Allocate the Build Area Data Sets	68
Exercise 2 Copy the Disassembler Sources to the Build Area Data Sets	69
Exercise 3 Create a Dimensions Project and Directories	70
Exercise 4 Set the Project and the Project Root	71
Exercise 5 Create the Items in Dimensions	72
Exercise 6 Add a Build Configuration to the Dimensions Project	75
Exercise 7 Add a Build Area to the Build Configuration	77
Exercise 8 Create Build Tools	79
Exercise 9 Create Build Option Groups	84
Exercise 10 Create Transition Rule Templates	87
Exercise 11 Create an Application Rule Template	92
Exercise 12 Create the Build Targets	94
Exercise 13 Check in the Build Configuration	104
Exercise 14 Run the Build	104
Exercise 15 Monitor the Build	108
Exercise 16 Verify that the Targets were Created in Dimensions	114
Exercise 17 View Derived Items	116
Exercise 18 View the Bill of Materials	118
Summary	121
Part 2 Overview of Dimensions Build	123
Chapter 4 About Dimensions Build	125
Introduction	126
What's New in Version 10.1.1?	127
Dimensions Build Terminology	127
The Build Flow	129
User Scenarios	131
Developer Builds	131
Project Builds	131
Continuous Developer Builds	132
Baseline Builds	132
Request Builds	133
Build Areas	133
Build Area Locations	133
Build Area Security and Ownership	134
Managed Development Areas	135
Build Areas and Active Item Revisions	135
Search Paths	135
Automatic Builds	136
Build Roles and Privileges	136

Capturing Build Outputs	136
Preservation Rules and Policies	137
External Item Revisions	137
Placeholder Item Revisions	138
Auditing Build Areas	138
Templates	138
Impacted Targets	138
Chapter 5	
Overview of the Build Administration User Interface	139
About Build Administration	140
Build Administration Main Window	140
Build Management Tab	141
Build Scheduling Tab	143
Build Monitoring Tab	145
Notification Tab	146
Part 3	
Configuring and Managing Builds	147
Chapter 6	
Configuring an MVS Build Environment	149
Pre-Requisites	150
Setting up your MVS Build Environment	150
Chapter 7	
Managing Build Settings	151
About Build Settings	152
Build Configuration Types	152
About Build Configuration Types	152
Viewing Build Configuration Types	154
Adding Build Configuration Types	154
Adding Options to Build Configuration Types	154
Modifying Build Configuration Types	155
Modifying Options for Build Configuration Types	155
Deleting Build Configuration Types	156
Deleting Options from Build Configuration Types	156
Build Tools	157
About Build Tools	157
About Build Options	157
Viewing Build Tools and Options	158
Adding Build Tools	159
Adding Options to Build Tools	159
Importing Build Tools	160
Modifying Build Tools	161
Modifying Build Tool Options	161
Deleting Build Tools	162
Deleting Build Tool Options	162
Build Option Groups	163
About Build Option Groups	163
About Build Options	164
Viewing Build Option Groups	164

Adding Build Option Groups	165
Adding Build Options to Build Option Groups	165
Modifying Build Option Groups	166
Modifying the Options of Build Option Groups	167
Deleting Build Option Groups	167
Deleting Options from Build Option Groups.	168
Transition Rule Templates.	169
About Transition Rule Templates.	169
About Build Options.	169
Viewing Transition Rule Templates	170
Adding Transition Rule Templates	170
Modifying Transition Rule Templates	171
Deleting Transition Rule Templates	172
Adding Input Masks to Transition Rule Templates	173
Modifying Transition Rule Template Input Masks	173
Deleting Input Masks from Transition Rule Templates	174
Adding Output Masks to Transition Rule Templates	174
Modifying Transition Rule Template Output Masks.	175
Deleting Output Masks from Transition Rule Templates	175
Adding Build Options to Transition Rule Templates	176
Modifying Build Options for Transition Rule Templates	177
Deleting Build Options from Transition Rule Templates	178
Adding Build Option Groups to Transition Rule Templates.	178
Deleting Build Option Groups from Transition Rule Templates.	179
Application Rule Templates	179
About Application Rule Templates	179
Viewing Application Rule Templates.	180
Adding Application Rule Templates	180
Modifying Application Rule Templates	180
Deleting Application Rule Templates	181
Adding Transition Rule Templates to Application Rule Templates.	181
Removing Transition Rule Templates from Application Rule Templates.	182
<i>Chapter 8</i>	
Managing Build Configurations	183
About Build Configurations	184
About Build Projects.	184
Basic Build Configuration Operations	186
Viewing Build Configurations.	186
Creating Build Configurations	186
Modifying Build Configurations	188
Copying Build Configurations	189
Deleting Build Configurations	190
Setting Build Configuration Properties	190
Build Areas	191
About Build Areas	191
Creating a New Work Area	192
Creating a New Deployment Area	193

Creating a New Library Cache Area	195
Modifying Build Areas	197
Deleting Build Areas	197
Sources	198
Adding a Build Source from a File	198
Using Wildcard Patterns to Add Build Sources from Dimensions	198
Adding Individual Files in Dimensions as Build Sources	200
Recursive Search	201
Modifying Build Sources	201
Deleting Build Sources	202
Build Targets	202
About Build Targets	202
Creating Build Targets	202
Modifying Build Targets	205
Deleting Build Targets	205
Inputs	206
About Inputs	206
Adding a New Target as an Input	206
Adding a New Source as an Input	206
Adding an Existing Target as an Input	207
Adding an Existing Source as an Input	208
Modifying Inputs	208
Deleting Inputs	209
Outputs	209
About Outputs	209
Scripts	210
About Scripts	210
Build Configuration Scripts	210
Transition Scripts	212
Reusing Scripts	212
Adding a Script	213
Viewing a Build Configuration Script	215
Viewing a Transition Script	216
Modifying a Build Configuration Script	216
Modifying a Transition Script	217
Deleting a Build Configuration Script	218
Deleting a Transition Script	218
Build Options	219
About Build Options	219
Creating Build Options	219
Modifying Build Options	221
Deleting Build Options	221
Creating Build Option Groups	222
Versions	222
About Versions	222
Exporting and Importing	222
About Exporting and Importing	222
Exporting a Build Configuration	223

	Importing a Build Configuration	223
	Importing an Ant build.xml File	224
	Importing an Openmake Build Configuration.	225
	Tips	225
<i>Chapter 9</i>	Executing Builds in the Build Administrator	227
	About Executing Builds	228
	Running a Single Build Configuration	228
	Running All Build Configurations in a Project	230
	Running a Build from a Scheduled Start	231
	Running a Non-Scheduled Build from the Build Scheduling Tab	231
	Running a Build Using Launch New Build	231
	Rebuilding Previously Executed Build Jobs	232
<i>Chapter 10</i>	Using Filters	235
	About Using Filters	236
	What Can Be Filtered	236
	Filtering Build Jobs by User Name	236
	Filtering Build Jobs by Date and Time.	238
	Filtering Dimensions Projects.	240
	Editing Filter Conditions	241
	Altering the Definition of a Filter Condition	242
	Removing Specific Filter Conditions	242
	Removing All Filter Conditions.	242
<i>Chapter 11</i>	Versioning	243
	About Versioning	244
	Example of Rollback	244
	Example of Rewrite	246
	Deleting Versions Not Allowed	247
<i>Chapter 12</i>	Scheduling Build Jobs	249
	About Scheduling.	250
	Managing the Scheduler Service	250
	Configuring the Scheduler Service	250
	Filtering Dimensions Projects.	252
	Managing Build Job Schedules	252
	Viewing Scheduled Build Jobs	252
	Adding Build Job Schedules	252
	Scheduling the Frequency and Range of Recurring Build Jobs	255
	Modifying Build Job Schedules	256
	Modifying the Frequency and Range of Recurring Build Jobs	258
	Deleting Build Job Schedules	259
	Checking the Results of Scheduled Build Jobs	259
	Running Builds from the Build Scheduling Tab.	259
<i>Chapter 13</i>	Monitoring Builds in the Build Administrator	261
	About Monitoring Builds	262

Reviewing Build Status Immediately After Launch	262
Viewing the Launch Attempt Log	264
Viewing the Build Job Details	265
Interpreting Build Monitor Events	265
Viewing Build Details	267
Monitoring a Running Build	268
Canceling a Running Build Job	269
Monitoring Past Builds	270
Viewing Build Execution History	270
Viewing Build Launch Attempts	271
Rebuilding the Selected Build Job	272
Viewing Dependencies	273
Viewing a Bill of Materials	275
Chapter 14	
Managing Notifications	277
About Notifications	278
Formatting Notification Templates and Events	278
Notification Templates	278
Notification Events	279
Notification Subscriptions	280
Managing the Notification Service	280
Configuring the Notification Service	280
Managing Notification Templates	281
Viewing Notification Templates	281
Adding Notification Templates	282
Editing Notification Templates	283
Copying Notification Templates	284
Deleting Notification Templates	284
Managing Notification Subscriptions	285
Filtering Dimensions Projects	285
Viewing Notification Events, Details, and Subscribers	285
Adding Notification Events	286
Editing Notification Events	287
Deleting Notification Events	288
Adding Notification Subscriptions	288
Editing Notification Subscriptions	289
Deleting Notification Subscriptions	290
Part 4	
Integrating Dimensions Build with Third Party Build Engines	291
Chapter 15	
Using Dimensions with Ant	293
About Using Dimensions Build with Ant	294
Overview	294
Deciding Where to Store the Ant Buildfile	294
Selecting or Creating a Dimensions Project	295
Creating or Selecting Build Areas	295

Importing Ant Buildfiles into a Build Configuration	295
Editing Details of the Imported Targets	297
Editing the Transition Script of an Imported Ant Target	298
Running the Imported Build Job.	299
Chapter 16 Using Dimensions Build with Openmake	301
About the Integration with Openmake	302
Templates.	302
Integration Steps.	303
Openmake Build Configuration Type.	304
Search Paths	304
Search Path Concatenation.	305
Active Item Revisions	306
Additional Search Locations	306
Importing TGT Files into a Build Configuration.	307
Producing Impact Analysis Data.	309
Return Codes	310
Part 5 Appendices	311
Appendix A Troubleshooting	313
About Troubleshooting	314
Viewing Errors Listed in Build Monitor Events	314
Specific Errors	315
Area <area name> is in use.	315
Attempt to close invalid connection	316
Failed to authenticate to the build agent	316
Failed to find product-specific upload rules.	316
No build areas found for the selected configuration	316
Template pbem_stop.cmd open failed	316
You do not have a role to extract item	316
Using Temporary Files to Debug Your Builds	317
Preserving Temporary Files	317
Locating the Temporary Files	317
The BRD File.	318
Appendix B The Dimensions Build Architecture	319
MVS Platforms.	319
Distributed Platforms	321
Appendix C Dimensions Build Utility Programs	323
The Primary Build Execution Monitor	324
Running the PBEM.	324
PBEM Parameters	325
PBEM Configuration Symbols	326
Example JCL for Running the PBEM	327
The Secondary Build Execution Monitor	327
SBEM Parameters	328

Example JCL for Starting the SBEM	329
Starting the SBEM from USS	330
MDHLLNK0	331
General Parameters	331
Output Control Parameters	331
Passthrough Control Parameters	332
Example JCL	334
bldcomms	336
Using bldcomms on MVS	337
Loading Multiple Members	337
Appendix D Dimensions Build Security	339
Phase 1: REXEC	340
Phase 2: dmlibsrv	340
Phase 3: The PBEM	340
Jobcards	340
Phase 4: The SBEM	341
Phase 5: Output Collection	341
Scheduled Builds	341
Dimensions and Build Area Accounts	341
The Default Userid	342
Dimensions Build Area Security	342
Using Build Areas Outside Dimensions Build	342
Index	345

Welcome to Serena Dimensions

Thank you for choosing Serena® Dimensions® CM, the configuration management component of Dimensions. Dimensions is a powerful process management and change control system that will revolutionize the way you develop software. Dimensions helps you organize, manage, and protect your software development projects on every level—from storing and tracking changes to individual files, to managing and monitoring an entire development cycle.

About Serena Dimensions	<p>Dimensions is a principal component of the integrated components that constitute the Serena Dimensions product. Other components include:</p> <ul style="list-style-type: none">■ Serena Dimensions RM, which offers full requirements management and traceability throughout the development lifecycle by centralizing and organizing requirements using role base views and a user configurable requirements process.■ Serena Command Center, which provides development project monitoring.
Purpose of this manual	<p>This manual describes how to use Dimensions Build, a build management, execution, and monitoring tool that is part of Dimensions CM.</p> <p>This manual has the following parts:</p> <ul style="list-style-type: none">■ Part 1 Quick Start <p>This part has quick start tutorials that introduce you to the basics of setting up build configurations and building applications in Dimensions Build. There are tutorials for C on Windows and Assembler on MVS (mainframe).</p> <ul style="list-style-type: none">■ Part 2 Overview of Dimensions Build <p>This part has an introduction to Dimensions Build concepts and an overview of the user interface.</p> <ul style="list-style-type: none">■ Part 3 Configuring and Managing Builds <p>This part describes how to use Dimensions Build including setting up build configurations, managing build settings, executing and monitoring builds, and setting up build schedules and notifications.</p> <ul style="list-style-type: none">■ Part 4 Integrating Dimensions Build with Third Party Build Engines <p>This part explains how to integrate Dimensions Build with Ant and Serena ChangeMan Builder (Openmake).</p> <ul style="list-style-type: none">■ Part 5 Appendices <p>This part contains information on troubleshooting, the Dimensions Build architecture, utility programs, and security.</p>
For more information	<p>Refer to the <i>Introduction to Dimensions CM</i> for a description of the Dimensions documentation set, a summary of the ways to work with Dimensions, and instructions for accessing the Online Help.</p>
Edition status	<p>The information in this guide applies to Release 10.1.1 of Serena® Dimensions® CM. Information concerning the Openmake product from Catalyst Systems, or the Serena ChangeMan Builder product, supersedes information that appeared in the earlier manual called <i>Serena ChangeMan Builder for Dimensions User's Guide</i>.</p>

Product Name Changes and New Terminology

Beginning with Dimensions 10.1, product name changes took place and terminology changes were also introduced.

The following table details the product rebranding:

Current product name	Legacy product name
Serena Command Center ^a	n/a
Serena Dimensions ^b	
Serena Dimensions Build ^c	n/a
Serena Dimensions	Serena ChangeMan Dimensions
Serena Dimensions RM	Serena RTM
Serena Mover	Serena ChangeMan Mover
Serena PVCS Version Manager	Serena ChangeMan Version Manager

a.New product.

b.Encompasses all Dimensions components.

c.New product.

The above product name changes have also led to changes in the Dimensions documentation set. Refer to the *Introduction to Dimensions* manual for a comprehensive list of the Dimensions documentation set.

The following table details the terminology changes:

Current terminology	Legacy terminology
custom list/request list	user list ^a
custom list/request list	custom list ^b
deployment area	build area
deployment stage	build stage
inbox	pending list
library cache area ^c	n/a
privilege ^c	n/a
project	workset
project working location	workset root directory
request	change document
user interface profile ^c	n/a
work area ^c	n/a

a.Desktop client.

b.Web client and Visual Studio integration.

c.New term. See the *User's Guide* and *Process Modeling Guide* for details of these new terms.

The latest Dimensions 10.1 documentation reflects the new terms, so if you are using a new Dimensions 10.1 component with a component from an earlier release of Dimensions (for example, the Migration Console in conjunction with Dimensions 9.1), you must bear in mind the terminology changes when working with the older components.



NOTE To ensure backward compatibility with existing customer applications, certain specialized features will continue to use legacy terminology, but the new terminology will be used when describing these features in the documentation. These features comprise:

- The Dimensions Command-Line Interface (dmcli).
- The public Dimensions C /C++ Developer's Toolkit and Java API (dmpmcli). This includes both the application interfaces and the manifest constants that come as part of this API.
- The Dimensions Data Interchange File Format (PDIFF).

Typographical Conventions

The following typographical conventions are used in the online manuals and online help. These typographical conventions are used to assist you when using the documentation; they are not meant to contradict or change any standard use of typographical conventions in the various product components or the host operating system.

italics	Introduces new terms that you may not be familiar with and occasionally indicates emphasis.
bold	Emphasizes important information and field names.
UPPERCASE	Indicates keys or key combinations that you can use. For example, press the Enter key.
monospace	Indicates syntax examples, values that you specify, or results that you receive.
<i>monospaced italics</i>	Indicates names that are placeholders for values you specify; for example, <i>filename</i> .
monospace bold	Indicates the results of an executed command.
vertical rule	Separates menus and their associated commands. For example, select File Copy means to select Copy from the File menu. Also, indicates mutually exclusive choices in a command syntax line.
brackets []	Indicates optional items. For example, in the following statement: SELECT [DISTINCT], DISTINCT is an optional keyword.
...	Indicates command arguments that can have more than one value.

Printing Manuals

As part of your Dimensions license agreement, you may print and distribute as many copies of the Dimensions manuals as needed *for your internal use, so long as you maintain all copies in strict confidence and take all reasonable steps necessary to ensure that the manuals are not made available or disclosed to anyone who is not authorized to access Dimensions under your Dimensions license agreement.*

Contacting Technical Support

Serena provides technical support for all registered users of this product, including limited installation support for the first 30 days. If you need support after that time, contact Serena Support at the following URL and follow the instructions:

<http://support.serena.com/>

Language-specific technical support is available during local business hours. For all other hours, technical support is provided in English.

The Serena Support Web page can also be used to:

- Report problems and ask questions.
- Obtain up-to-date technical support information, including that shared by our customers via the Web, automatic E-mail notification, newsgroups, and regional user groups.
- Access a knowledge base, which contains how-to information and allows you to search on keywords for technical bulletins.
- Download fix releases for your Serena products.

Part 1

Quick Start

Part 1: Quick Start contains the following chapters

C Quick Start, Part One	19
C Quick Start, Part Two	49
Assembler MVS Quick Start	65

Chapter 1

C Quick Start, Part One

To help you get started using Dimensions, this chapter walks you through the process of setting up and building a simple "Hello, World" program in C.



NOTE Mainframe users should go directly to the [Assembler MVS Quick Start](#) on page 65.

Before You Begin	20
Create a New Work Area	20
Display the Dimensions Build Administration Console	22
Create a Build Configuration	23
Attach a Build Area	26
Create the Source File	27
Add a Build Target	28
Check in the Build Configuration	35
Run the Build	36
View the Execution Details of the Build	40
Re-defining Build Options	42
Create A Build Configuration Using Copy	44
Summary	46
Preview of Other Features	47
Additional Information	47

Before You Begin

Before you start the tutorial, check the following pre-requisites:

- You have installed, configured, and licensed a Dimensions 10.x server and clients on the host machine. For details, see the *Installation Guide* and the *Tool Administration Guide*.
- You have access to a user ID with Administrator privileges (such as "dmsys").
- You have set up Dimensions users and groups.
- The Serena Dimensions Listener service, Oracle TNS Listener service, and Serena License Server are all running on the host machine. To check if these services are started, choose Programs | Control Panels | Administrative Tools | Services. For all services the Status column should be Started. If a service is stopped, right-click it and select Start.
- You have a C/C++ compiler installed and that your PATH environment variable is set accordingly. The exercises use the GNU compilers.
- You have set up a Dimensions item library. This is necessary in order to have the build result added as a Dimensions item.

Exercise 1 Create a New Work Area

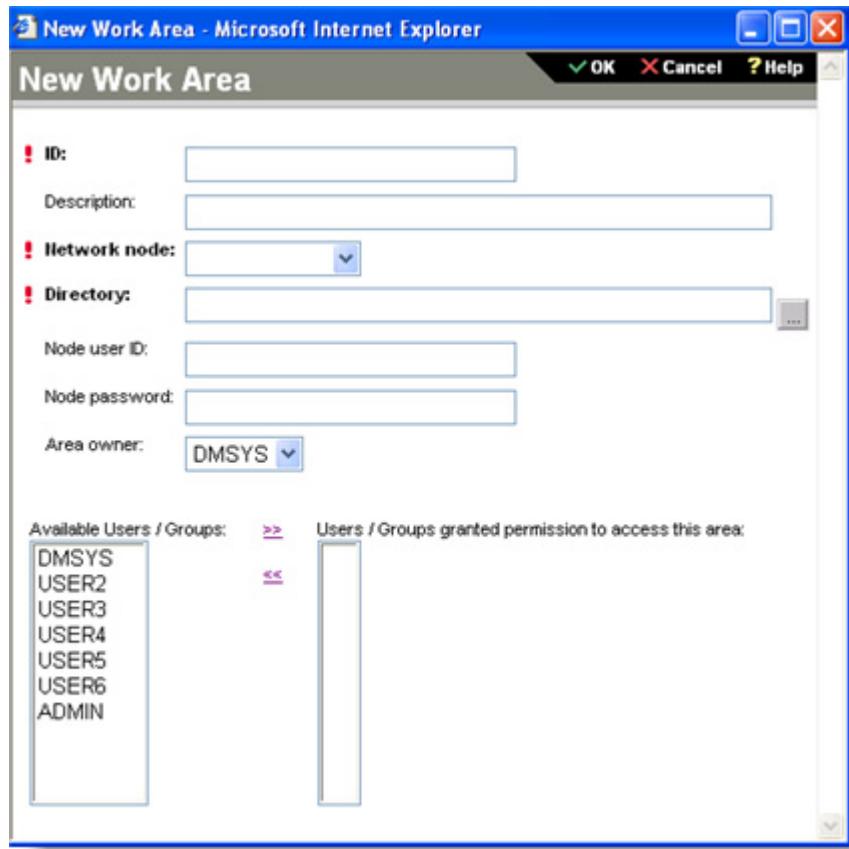
Although you can create build areas during the process of defining a build project, it is convenient to create the build areas first.

Build areas appear in Dimensions under build configurations, but they are created in the Dimensions Administration Console, and are referred to there as file areas.

To create a new work area:

- 1 Log in to the Dimensions Administration Console.
- 2 Under the Distributed Development heading, click on **Area Definitions**. The Area Definitions page appears.
- 3 Click the New icon. The pop-up menu displays the types of area that you can create:
 - Work area
 - Deployment area
 - Library cache area

- 4 Select **Work Area**. The New Work Area dialog box appears:



- 5 Fill out the fields as follows:

- **ID**

The identifier for the work area; for example, "HelloProject_WorkArea".

- **Description**

Enter a description for the work area.

- **Network Node**

The name of the machine that Dimensions is to use when executing the build.

- **Directory**

The name of the directory that Dimensions is to use as the base directory; relative pathnames will be evaluated based on this directory.

- **Node User ID**

The user ID that Dimensions is to use when executing the build.



NOTE On Windows 200x platforms, the user should have administrator privileges. On Windows XP, UNIX, USS, or MVS, these privileges are not required.

- **Node Password**

The user password that Dimensions is to use when executing the build.

- **Area Owner**

This is the user ID that has permission to edit the build area.

- **Available Users/Groups and Users/Groups granted permission to access this area**

Select entries from the Available Users/Groups. Use the >> and << links to move the selected entries to the list of users/groups with permission to access the area.

- 6 When you have finished filling out the dialog box, click **OK**. A Results dialog box briefly confirms the creation of the new area. The new area also appears in the File Areas List.
- 7 Click Home to return to the home page of the Dimensions Administration Console.

Exercise 2 Display the Dimensions Build Administration Console

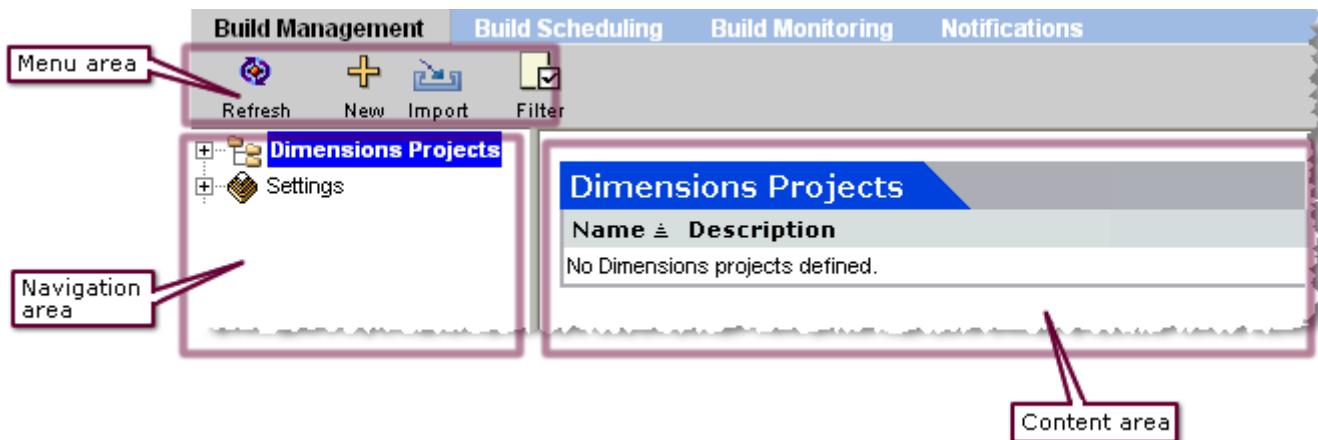
After build areas have been created, you are ready to work in Dimensions itself. The first step is to display the administration console.

To display the Dimensions administration page:

- 1 From the Dimensions Administration Console, click **Build administration**. The Dimensions administration page appears.



In this document we will use the following names to refer to each area:



Notice the four tabs just underneath the application title:

- **Build Management**

The controls on this tab affect Dimensions projects, build configuration, build areas, and build settings. Most of the configuration work for Dimensions is done on this tab.

- **Build Scheduling**

The controls on this tab affect the scheduling of regular or delayed builds.

- **Build Monitoring**

The controls on this tab allow you to observe the details of running build jobs, examine the history of previously-executed build jobs, and initiate rebuilds.

- **Notifications**

The controls on this tab allow you to define templates for notification messages (such as "Build Completed") and to define subscriptions to messages for specific build events.

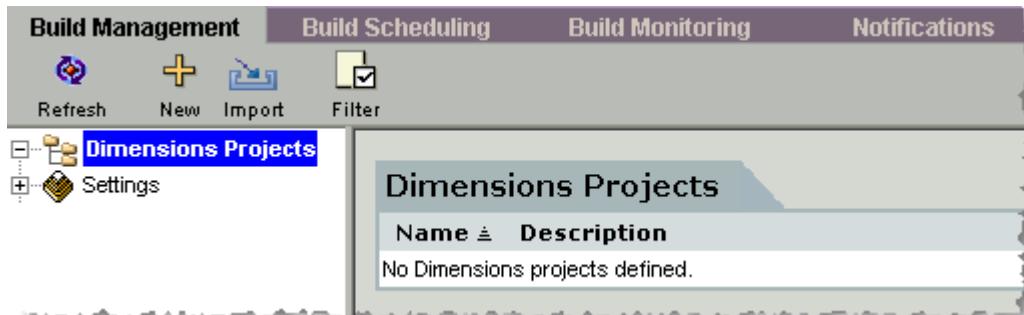
Exercise 3 Create a Build Configuration

When you first arrive at the administration page, you see two entries in the tree on the left-hand side of the Build Management page: Dimensions Projects and Settings.

To contain the details of your build job, you create a new build configuration. The process of creating a new build configuration requires you to select a Dimensions project. The project you select then appears in the tree of Dimensions projects.

To create a new build configuration:

- In the navigation area, select Dimensions Projects if it is not selected already. The **Dimensions Projects** heading appears.



- Click the **New** icon (shaped like a plus sign):



This displays the Add New Build Configuration dialog box.

Properties		Pre-Script	Main Script	Post-Script	Clean-Up Script
<input checked="" type="checkbox"/> Parent Project:	\$GENERIC:\$GLOBAL				
! Name:					
Description:					
Project Relative Path:					
Project for Targets:	the same as Parent Project				
Platform:	Win32				
Launch Timeout:	Default				
Execution Timeout:	Default				
Type:	Default				

The Properties tab of this dialog box, displayed by default, contains the identifying details of the build configuration.



NOTE The other four tabs in this dialog box contain details for various types of scripts. In this exercise, you will use only the Properties tab.

- 3 Complete the editable Properties tab fields as follows:

- **Parent Project**

Select a Dimensions project from the list. You might want to select a project other than \$GENERIC:\$GLOBAL since that project may not have upload rules or item libraries properly defined yet.

- **Name**

Enter a name for the build configuration.

- **Description**

Enter a description for the build configuration.

- **Project Relative Path**

Leave this blank. This represents the base directory of the location where you will build. That location is defined using the work area you created earlier.

- **Project for Targets**

This determines where the products of the build will be uploaded after completion. Accept the default choice of **the same as Parent Project**, or select a different project from the list.

- **Platform**

Enter the appropriate platform (Win32, AIX, HPUX, and so on).

- **Launch Timeout**

Leave this field as Default. (The field represents the number of seconds that Dimensions will wait before abandoning a launch attempt.)

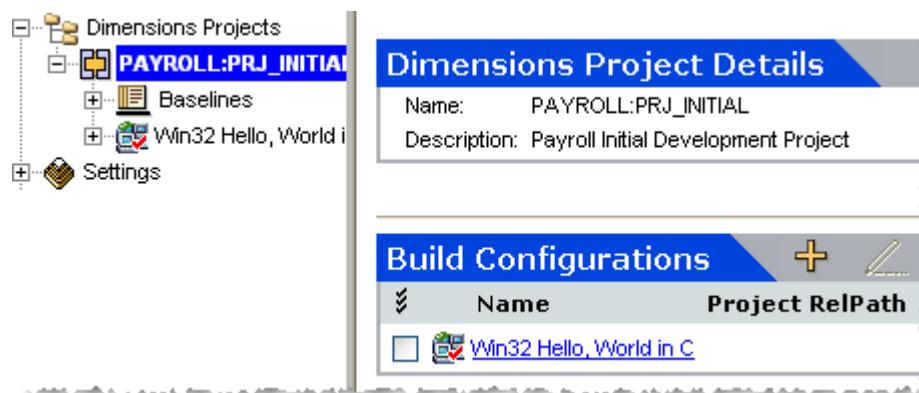
- **Execution Timeout**

Leave this field as Default. (The field represents the number of seconds that Dimensions will wait before abandoning an execution attempt.)

- **Type** (Default or Openmake)

Leave this field as Default. Using Dimensions with Openmake is discussed in [Using Dimensions Build with Openmake on page 301](#).

- 4** Click **OK**. A Results dialog confirms the creation of the build configuration. The name of the build configuration also appears in the Build Configurations area.



Notice that the icon next to the build configuration has a red check mark.



This indicates that the build configuration is checked out. You must check in the build configuration before Dimensions can use it. (You will do this later in the tutorial.)

- 5** Select the name of the newly-created build configuration. The content area of Dimensions displays sections for build areas, build targets, and build options.

Exercise 4 Attach a Build Area

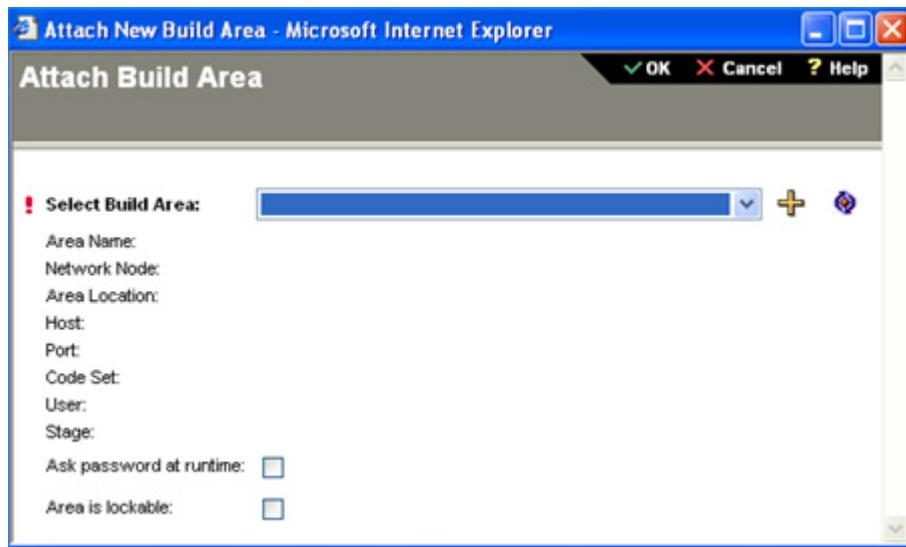
The build area defines a starting point for the build job activity. You specify the location of source files relative to this area, as well as the location of the final build products.

Be sure that the build configuration is checked out before attempting to make changes.

To attach a build area:

- From the Build Management tab, select the name of the build configuration if it is not selected already. You should be able to see the Build Areas section in the content area.

- 2 Click the New object icon (shaped like a plus sign) next to the Build Areas section title. The Attach Build Area dialog box appears.



- 3 Fill out the editable fields as follows:

- Select Build Area

The list displays the Dimensions build areas that are appropriate for the lifecycle state specified in the Stage field. Select the desired build area.

- Ask password at runtime

This means that Dimensions will prompt you for the OS User password each time the build is executed. This is true whether or not you specify the password ahead of time.

- Area is lockable

This means that the build area can be locked so that concurrent builds cannot occur. If you are worried about another build interfering with the contents of the build area, then select this check box.

- 4 Click OK. A Results dialog briefly confirms the successful selection of the build area.

Exercise 5 Create the Source File

For the purpose of this Quick Start, this exercise describes the creation of a source file to use as an input to a build target.

The following steps describe the creation of a simple "Hello, World" program to use as the source file for the build.

To create the source file:

If you do not already have a "Hello, World" file in C, you can copy the following C source:

```
int main(int argc, char **argv)
{
    printf("\nHello, World!\n");
    return(0);
}
```

- 1** Copy the above text and paste it into a file.
- 2** Save the file as `hello.c`, in the work area you defined earlier.
- 3** Do a test compile from the command-line to make sure your program works.
- 4** Check the file into Dimensions, adding it to the `src` folder of the project you specified earlier.

This source file will be used as an input to the build target.

Exercise 6 Add a Build Target

A build target is the desired product of the build job. Adding a build target involves defining the build target as well as specifying what sources are used to create the build target.

Targets can be real or virtual:

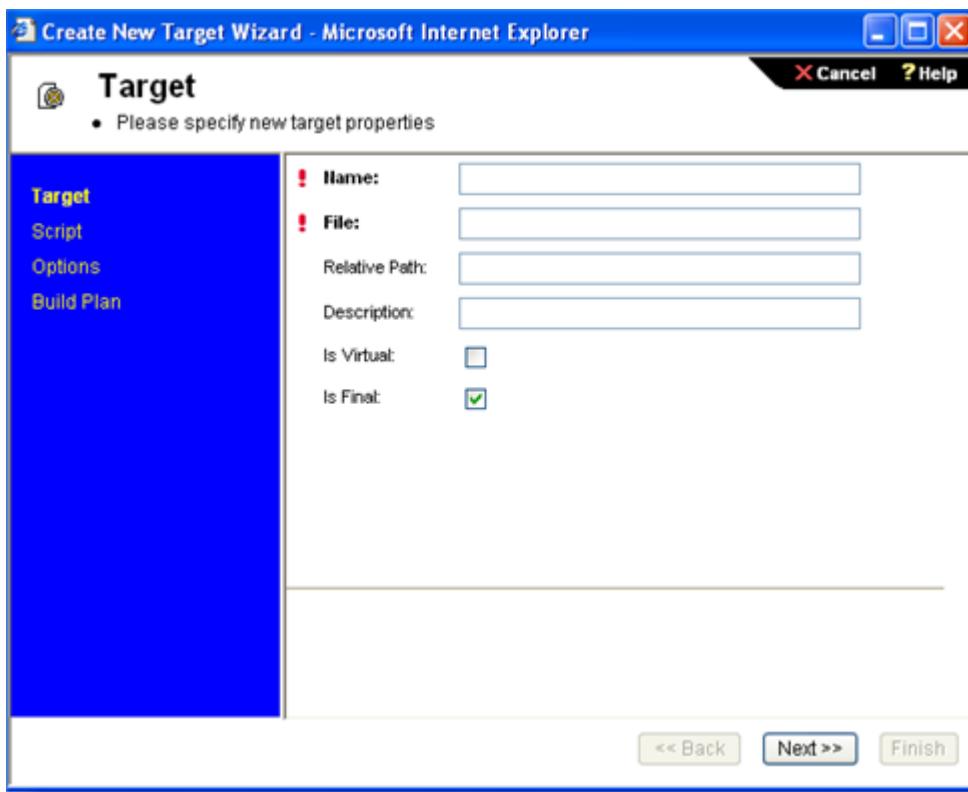
- An example of a real target would be an executable file or library.
- An example of a virtual target would be "all". Such a target would cause the rebuilding of all associated modules, but there would not be a file at the end named "all".

Be sure that the build configuration is checked out before attempting to make changes.

To define a build target:

- 1** From the Build Management tab, select the name of the build configuration. You should now be able to see the Build Targets section in the content area.

- 2** Click the New object icon (shaped like a plus sign) next to the Build Targets section title. The Create New Target wizard appears.



- 3** Notice that the left side of the wizard has four selections: Target, Script, Options, and Build Plan. By default, the Target page displays.

Fill out the fields on the Target page

The Target page specifies the name of the build target and the path to it.

- 1** Fill out the fields as follows:

- **Name**

Enter a descriptive name for the build target.

- **File**

Enter the actual filename of the build target. For this exercise, enter hello.exe.

- **Relative Path**

This is the path to the build target, relative to the build area directory.

- **Description**

Enter a description for the build target. This can be a longer description than the one entered in the Name field.

- **Is Virtual**

If checked, indicates that the target is a virtual target (such as "all").

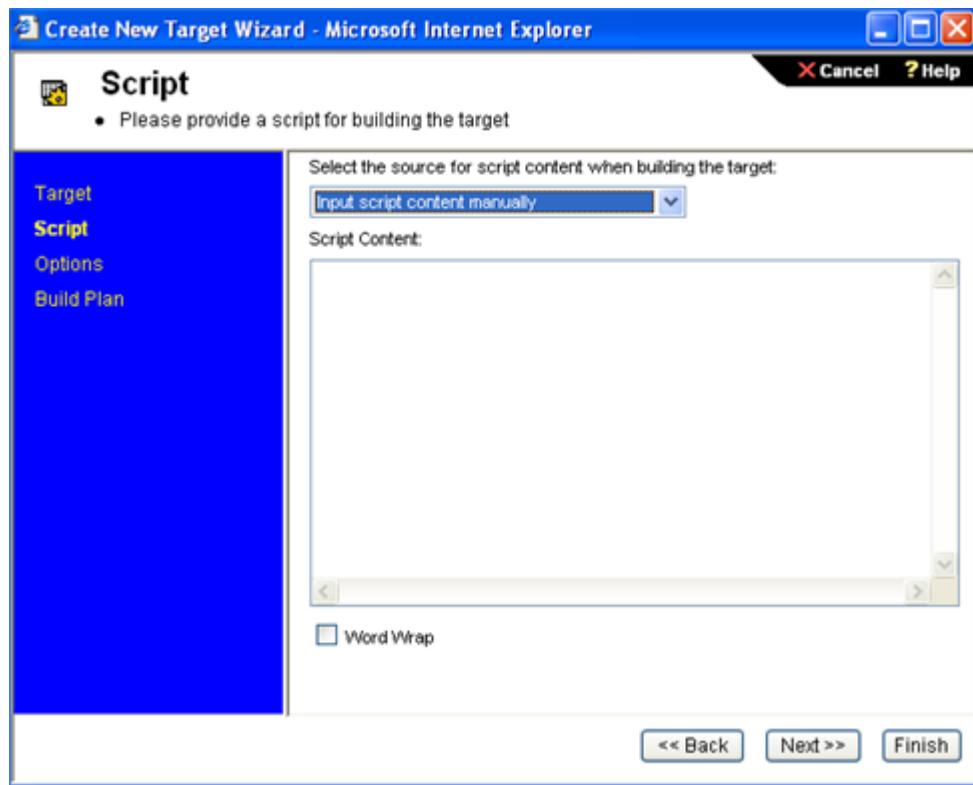
- **Is Final**

If checked, indicates that the target is a final target and not an intermediate target; that is, the target will not be used as the input to a subsequent build operation.

- 2 When you are done with the Target page, click **Next**. The Script page appears.

Fill out the fields on the Script page

The Script page determines the values for the transition script. This script specifies the details of how the target is to be created.



To create a build script

- 1 Fill out the fields as follows:

- Select the source for script content when building the target

Use the drop-down menu to select the source for the script. The choices are:

- **input script content manually**
- **use Dimensions file as a script**
- **use a file in the build area as a script**
- **Use the build configuration**

For this exercise, select **input script content manually** and enter the following command to compile your C program.

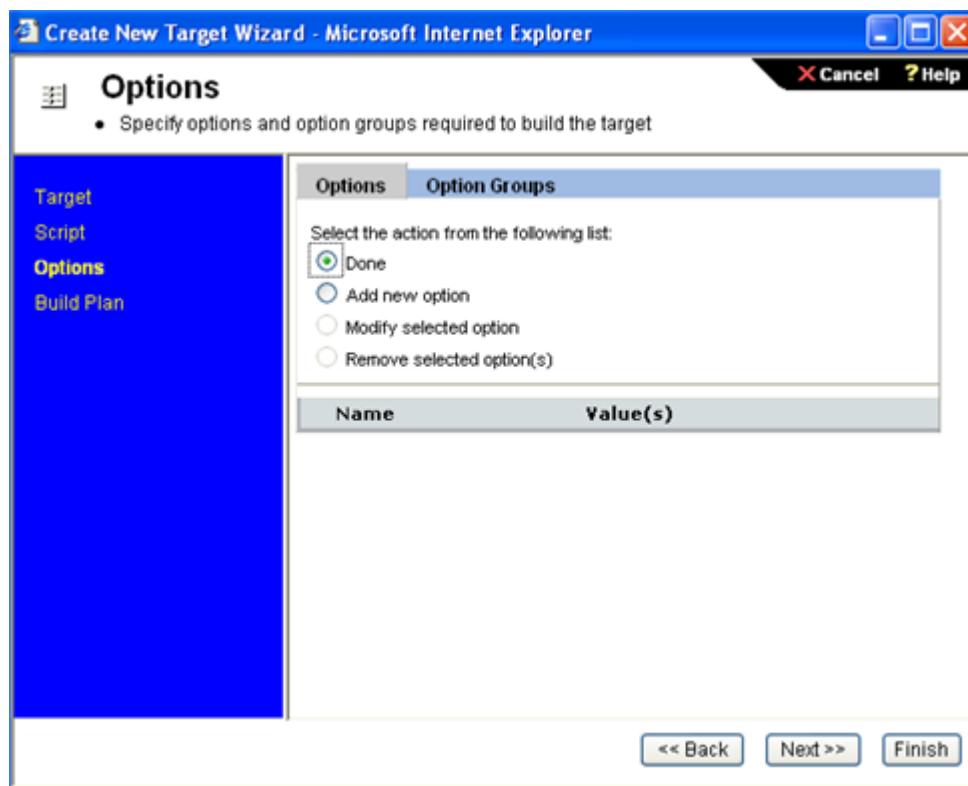
```
gcc -o %OutputFileName. hello.c
```

OutputFileName is a build option that you will define in a later exercise.

- 2 If you wish to have the contents of the Script Content area wrap, check the box titled Word wrap.
- 3 When you are done with the Script page, click **Next**. The Options page appears.

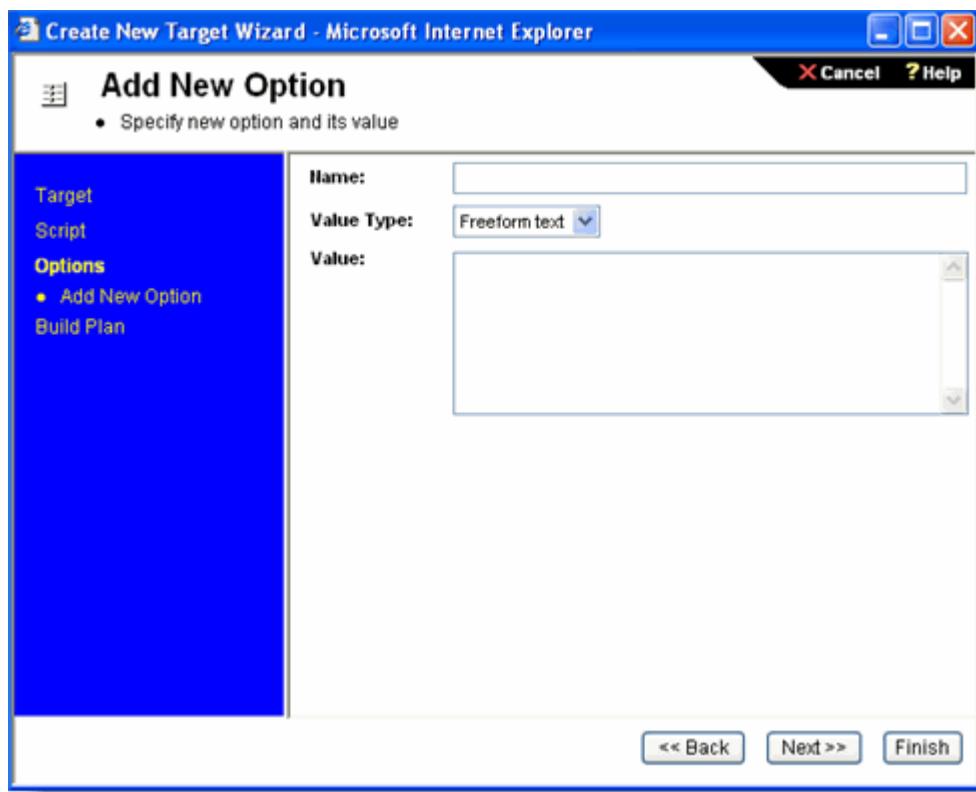
Fill out the fields on the Options page

The Options page allows you to define build options and option groups that may be useful or necessary while executing the build job.



To add a new option to a target:

- 1 Click **Add new option**, then click Next. The Add New Option page appears.



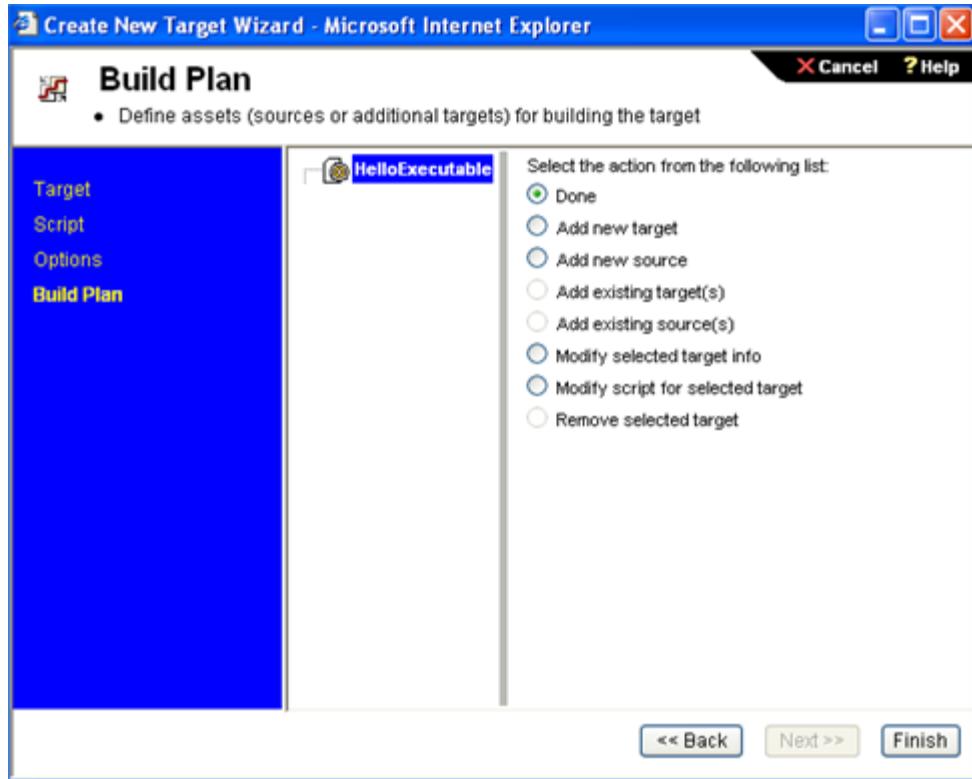
- 2 For Name, enter OutputFileName.
- 3 For Value Type, accept the default of Freeform text.
- 4 For Value, enter the following:

Hello

This defines the OutputFileName option used in an earlier exercise.
- 5 Click **Next**. You are returned to the Options page. The new option appears in a table on the page.
- 6 Click Next again to move to the Build Plan page.

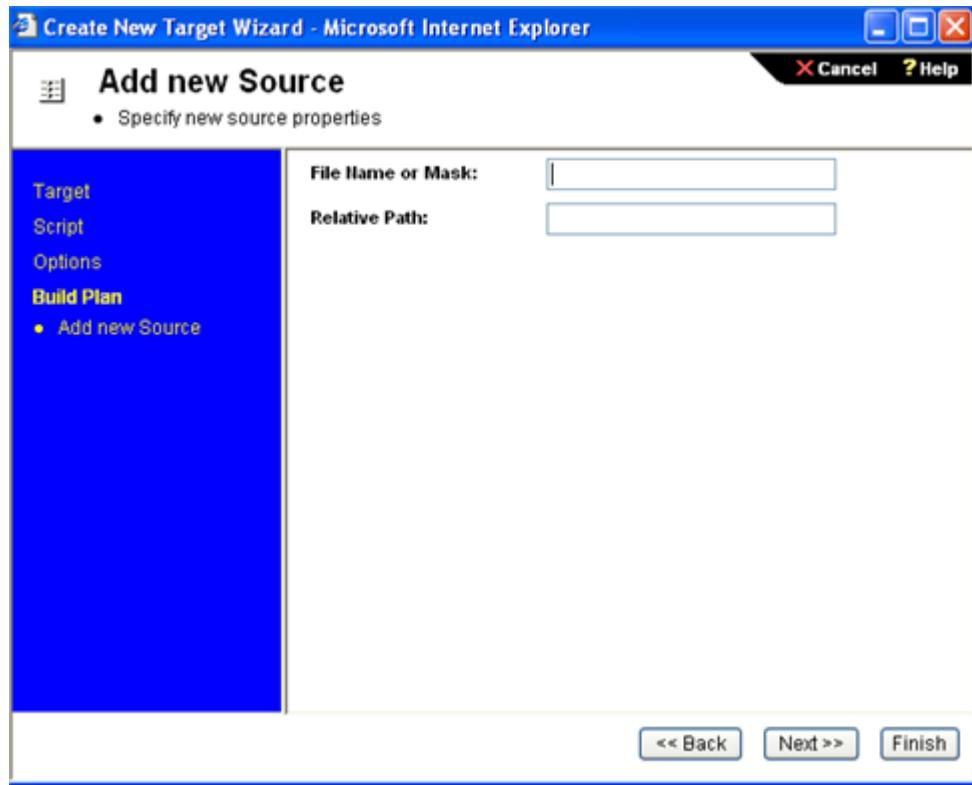
Fill out the fields on the Build Plan page

The Build Plan page displays a set of radio buttons that offer you a chance to add source files or targets to the build.

**To add a new source file:**

- 1 On the Build Plan page of the Create New Target Wizard, click the **Add new source** radio button.

- 2** Click **Next**. The Add new Source page appears.

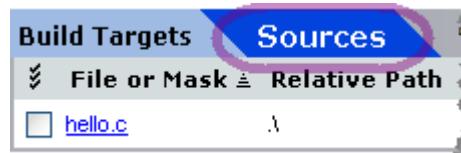


- 3** For **File name or mask**, enter hello.c.
- 4** For **Relative Path**, enter the path to the file, relative to the work area. Since you added hello.c to the src folder earlier, enter \src.
- 5** Click **Next**. The Build Plan page reappears, and the new source file appears in the tree display for the Build Plan page.
- 6** Click **Finish** to complete the target creation process.

A Results dialog box briefly displays a confirmation that the new target was created, and the new target appears in the Build Targets list.



By clicking the Sources section title, you can also see a list of the source files associated with the target.



Note that the build option you defined does not appear after you close the Create Target wizard. It has been created, but is associated with the build target, so you must select the build target in order to see it.

Exercise 7 Check in the Build Configuration

After completing your edits on the build configuration, it is time to check in the build configuration. Otherwise, Dimensions will not be able to execute the build job.

To check in the build configuration:

- 1 From the navigation area of the Build Management tab, select the name of the build configuration. The icon to the left of the build configuration's name should have a red check mark.
- 2 Click the **Check In** icon:



The Provide check-in comment dialog box appears.

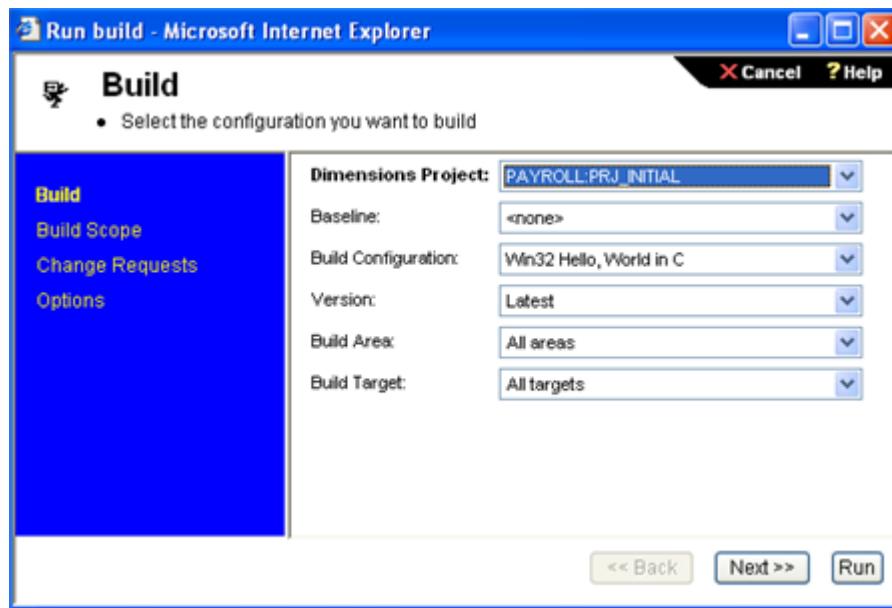
- 3 Enter a comment describing the changes, then click **OK**. A Results dialog box briefly displays a confirmation that the build configuration was checked in. Also, the icon next to the build configuration no longer includes a check mark.

Exercise 8 Run the Build

Once you have a checked-in build configuration, you can run the build job.

To run the build job:

- 1 From the navigation area of the Build Management tab, select the name of the build configuration if it is not selected already. The **Run** icon appears.
- 2 Click the Run icon. The Run Build wizard appears.



- 3 On the Build page, select the appropriate choices for each field:

- Dimensions Project

By default, the Dimensions project containing the build configuration that you selected appears. (If you try to select a build project that contains no build configurations, Dimensions displays an error message.)

- Baseline

This field appears only if the Dimensions project in the first field contains one or more baselines.

- Build Configuration

By default, the build configuration that you selected appears.

- Version

Latest executes the latest checked-in version. Other versions, if available, appear in the list.

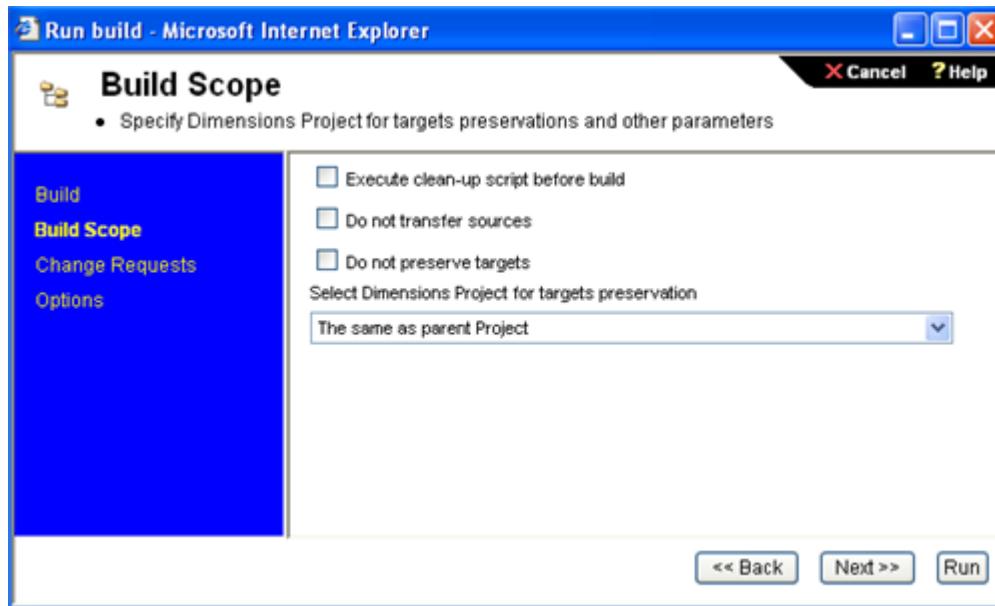
- Build Area

Other attached build areas, if available, appear in the list.

- Build Target

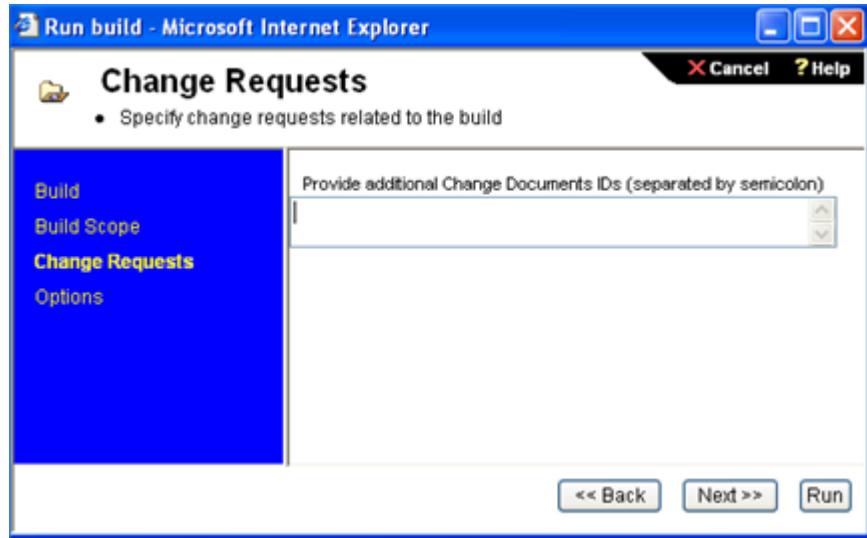
Other targets defined for the build configuration, if available, appear in the list.

- 4 Click **Next**. The Build Scope page appears.



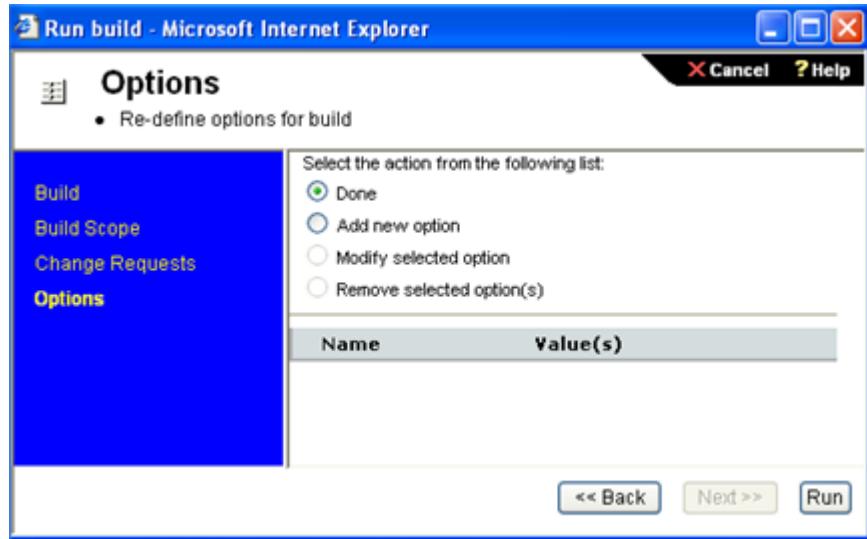
- 5 On the Build Scope page, select the appropriate choices for each of the check box fields:
- Checking the box labeled **Execute clean-up script before build** will cause Dimensions to execute the Clean-Up Script defined in the build configuration. This is most commonly used to delete unwanted items from the build area. The script is executed before source files are transferred into the build area. Note that if no script is defined, checking this box does nothing.
 - Checking the box labeled **Do not transfer sources** will cause Dimensions to not retrieve source files from Dimensions into the build area.
 - Checking the box labeled **Do not preserve targets** will cause Dimensions to not preserve the build targets by uploading them into Dimensions. Checking this box causes the Select Dimensions Project list to disappear.
- 6 If you accept the default choice of preserving targets, use the **Select Dimensions Project for targets preservation** drop-down menu to choose the project that should receive the targets.

- 7 Click **Next**. The Change Requests page appears.



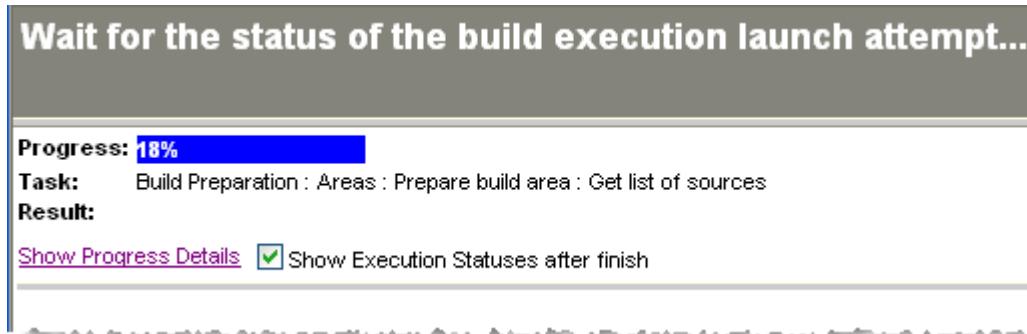
If there were change requests associated with the Dimensions project, you would be able to select them here, or enter a specific change request yourself.

- 8 As there is no need to associate a change request right now, click **Next**. The Options page appears.



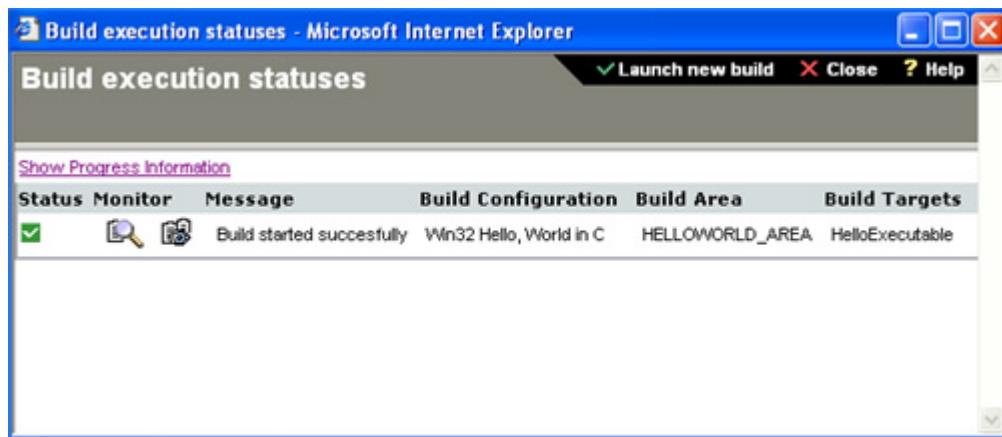
In a later exercise, you will use this page to redefine an existing build option.

- 9 Click **Run**. The wizard page displays the **Wait for the status of the build execution launch attempt** dialog box.



A progress bar shows the percentage of launch activity that is complete. When the progress bar reaches 100%, the build is ready to begin execution.

- 10 Eventually, the page displays the message "Build execution statuses", and the status of your build job appears:



Exercise 9 View the Execution Details of the Build

You can view the execution details of the build using the icons on the Build execution status dialog box. The two icons under the Monitor heading provide ways to do this.

Launch Log

- From the Build execution statuses dialog box, click this icon to display the Launch Log.



The Launch Log shows the details of the attempt to launch the build.

 A screenshot of a Microsoft Internet Explorer window titled "View launch attempt log". The content area displays a command-line log of a build attempt. The log includes file transfer statuses, command execution details, and template processing information.

```

File transfer statuses:
SUCCESS: File src/hello.c already exists

Passing the template for execution
cmd = cmd /c C:\DOCUME~1\dmssys\LOCALS~1\Temp\TPL153~1.BAT
>"C:\DOCUME~1\dmssys\LOCALS~1\Temp\tp12530-2.tmp" 2>&1
Template processing and submission complete: no errors

BRD file name = C:\DOCUME~1\dmssys\LOCALS~1\Temp\pt5301.tmp
  
```

If you have a problem that occurs during the attempt to launch the build—that is, while files are being transferred into the build area, and while the script to build the target is being evaluated—the launch attempt log can provide useful clues.

- Click OK to close the Launch Log.

Build Job Details

- From the Build execution statuses dialog box, click this icon to display the Build Job Details.



Build Job Details show information such as the start time and configuration used, as well as the build monitor events:

Build Job Details

Build Job Details

Build Job ID:	304
Started at:	Fri Aug-11 11:05AM
Finished at:	Fri Aug-11 11:05AM
Initiated by:	dmsys
Build Result:	Build job executed successfully
Dimensions Project:	PAYROLL:PRJ_INITIAL
Build Configuration:	Win32 Hello, World in C
Platform:	Win32
Targets:	HelloExecutable;

Build Monitor Events

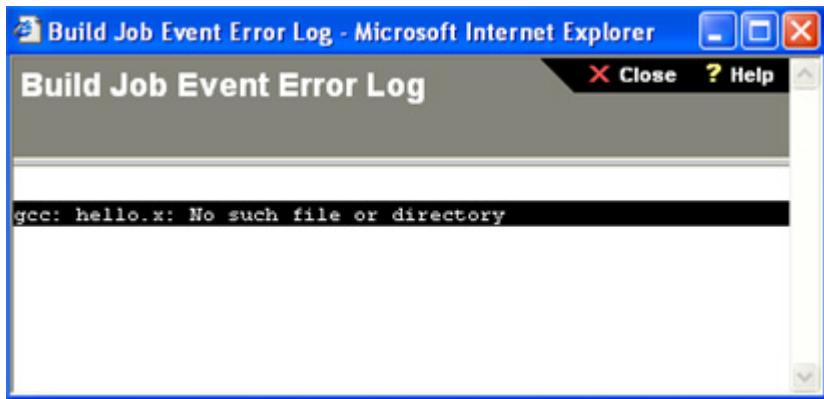
Order	Step	Type	Time	Target
1		Build started	Fri Aug-11 11:05AM	
2		Target Build started	Fri Aug-11 11:05AM	hello.exe
3		Target Build finished	Fri Aug-11 11:05AM	hello.exe
4		Information	Fri Aug-11 11:06AM	hello.exe
5		Build finished	Fri Aug-11 11:05AM	

If errors occur, the Build Monitor Events will display a red icon for each step that contains an error:

Build Monitor Events

Order	Step	Type	Time	Target
1		Build started	Fri Aug-11 10:57AM	
2		Target Build started	Fri Aug-11 10:57AM	hello.exe
3		Target Build finished	Fri Aug-11 10:57AM	hello.exe
4		Build finished	Fri Aug-11 10:57AM	

If you click the red icon, Dimensions displays an error log:



- 2 Click OK to close the Build Job Details.

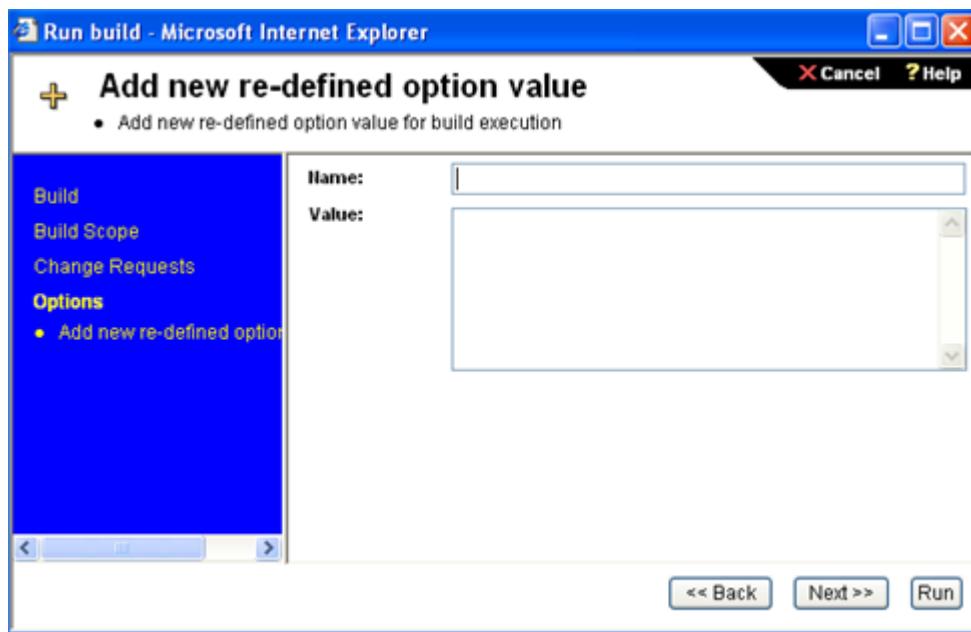
Exercise 10 Re-defining Build Options

In an earlier exercise, you defined an option called OutputFileName to specify the name of the output file. That build option was saved when the build configuration was checked in.

The Options page in the Run Build wizard page is different. It is designed to allow you to redefine existing options. The option definitions from this page will be in effect only for a single build execution. In this exercise, you will redefine an option to demonstrate this.

- 1 Select the same build configuration you created in earlier exercises.
- 2 Click **Run**. The Run Build wizard appears.
- 3 Click **Options**. The Options page appears.

- 4** Click **Add new option**, then click Next. The Add New Re-defined Option page appears.



- 5** For **Name**, enter OutputFileName. This is the same name as the option you defined as part of the build target.
- 6** For **Value**, enter the following:
HelloAgain
This redefines the output file name to HelloAgain instead of Hello.
- 7** Click Next. The Options page reappears, with the newly-redefined option listed.
- 8** Click **Run**. The launch attempt status and build execution status dialog boxes appear.
- 9** Leave the build execution status dialog box up, then use an operating system explorer or command prompt to view the build area and inspect the output of the build job. The output file executable HelloAgain is created.
- 10** Return to the Build execution statuses dialog box and click **Launch new build**. The Run Build wizard reappears.
- 11** Click **Options**. The Options page appears, and this time the redefined option is gone.
- 12** Click **Run**. When the build finishes execution, inspect the build area again. The output file has been created, but under the name specified in the original definition of the build option. This demonstrates that the build option from the Run Build wizard was only temporary.

Exercise 11 Create A Build Configuration Using Copy

If you have created one build configuration, and wish to create another, similar, build configuration, you can use Copy to save yourself some steps.

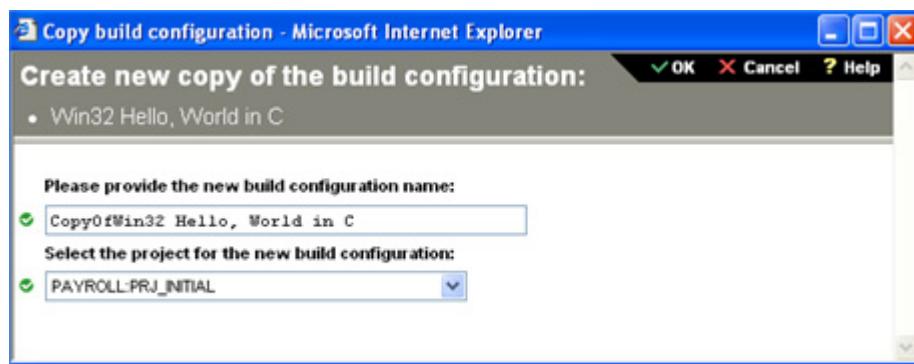
Copy the Existing Build Configuration

To copy an existing build configuration:

- 1 From the Build Management tab, select the name of the build configuration.
- 2 Click the **Copy** icon.



The **Create new copy of the build configuration** dialog box appears:



- 3 Provide a new name for the build configuration (or accept the default).
- 4 Select the Dimensions project that should receive the new build configuration. The default value is the same project.
- 5 Click **OK**. The new build configuration appears in the list of build configurations.

Edit the New Build Configuration

After creating a copy of a build configuration, of course, you will need to make a few changes. For this example, you will edit the build configuration from a C source file to a C++ source file, and also edit the transition script accordingly.

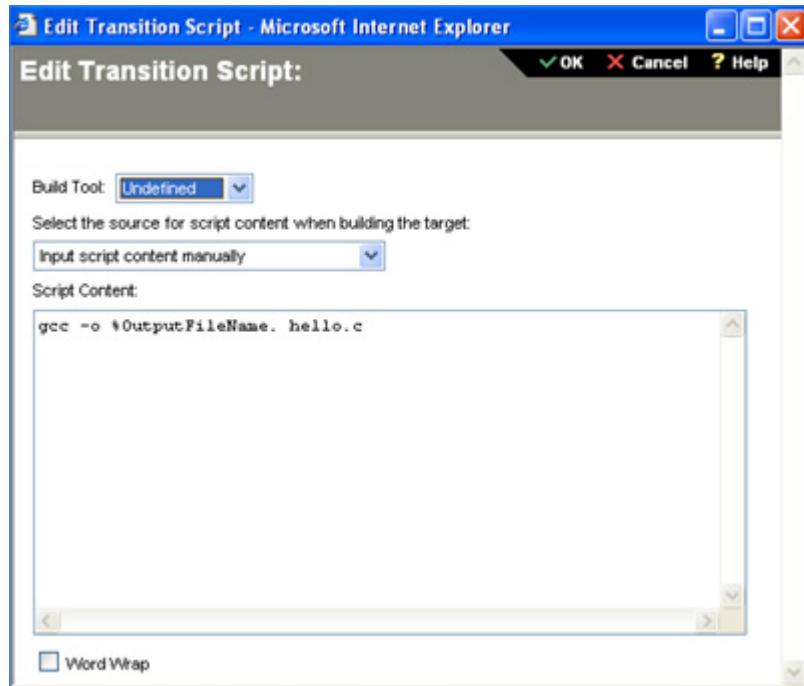
To edit the new build configuration:

- 1 Click the name of the new build configuration so that you can see the underlying components, such as Build Areas and Build Targets.

- 2** Click the name of the build target. You should now be able to see the Transition Details heading:



- 3** Click the pencil icon next to the Transition Details section title. The Edit Transition Script dialog box appears:



- 4** Replace the existing text with text appropriate to the compile of a C++ program; for example:

```
g++ %OutputFileName. hello.cc
```

- 5** Click **OK**. A Results dialog box briefly confirms the edit.

- 6** With the name of the target still selected, you should be able to see the Inputs tab.



- 7** Under the Inputs tab, click the link corresponding to the existing C source file. The Edit Source dialog box appears.
- 8** Edit the file name or mask so that it contains the name of a C++ source file named `hello.cc`.
- 9** Click **OK**. A Results dialog box briefly confirms the edit.

- 10** Finally, use a text editor to create the `hello.cc` source file. Be sure to create this file in the work area:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "\nHello, World!\n";
}
```

Check in the New Build Configuration

After you have finished the edits on the new build configuration, remember to check it in. Otherwise, Dimensions will not be able to execute the build job described by the build configuration.

Run the New Build Configuration

To execute the new build configuration, follow the steps described in [Run the Build](#) on page 36.

Summary

In this Quick Start you learned the following:

- How to create a work area
- How to use the Build Management page of the Dimensions Build Administration console
- How to create a build configuration
- How to attach a work area to a build configuration
- How to define a build target
- How to create a build script
- How to create a build option
- How to temporarily redefine a build option
- How to add a source file to a target definition
- How to check in a build configuration
- How to run a build
- How to view the execution details of a build
- How to create a build configuration using copying

What's Next

Distributed-platform users should go on to the next quick start, [C Quick Start, Part Two](#), which demonstrates how to use Dimensions with sources in Dimensions, and demonstrates the use of wildcard patterns when adding multiple source files.

Preview of Other Features

Dimensions contains many features and this chapter was just an introduction. In subsequent chapters, you will learn about many other features, such as:

- Build Job Scheduling
- Build Job Monitoring
- Notifications
- Importing and exporting build configurations
- Importing Openmake TGT files
- Importing makefiles

Additional Information

Following is some additional information about build areas.

More About Build Areas

There are three types of file areas/build areas:

- Work areas

A work area is defined for an individual developer, or for a group, so that file operations such as check in and check out automatically use the work area.

- Deployment areas

A deployment area is associated with a particular Dimensions global lifecycle stage. When an item file is deployed to a particular stage, Dimensions automatically places a copy of the item in the associated deployment area. For example, when an item reaches the SYSTEM TEST stage, Dimensions can place a copy of the item into a deployment area reserved for system test builds.



NOTE Deployment scripts should be placed in the directory \$DM_ROOT/templates on the node that hosts the area. Do not place the deployment scripts in the area itself.

- Library cache areas

A library cache area increases performance for sites that frequently have to retrieve items from remote or slow connections. You should not have to administer or configure these areas while using Dimensions.

This is only a brief introduction to build areas. To learn more, read the File Areas Definitions chapter of the *Serena Dimensions Process Modeling User's Guide*.

Chapter 2

C Quick Start, Part Two

This chapter picks up where the previous chapter, "C Quick Start, Part One," left off. You will build the Payroll example that comes with Dimensions, and check the resulting build products into Dimensions.



NOTE Mainframe users should go directly to the [Assembler MVS Quick Start](#) on page 65.

Objectives	50
Before You Begin	50
Create a New Work Area	50
Add the Payroll Project Build Configuration	52
Attach the Build Area	53
Define the Build Target	54
Define the Build Script	55
Add Files from Dimensions	56
Add Existing Sources as Inputs	58
Check in the Build Configuration	59
Run the Build	60
Verify that Target Was Created	62
Verify that Target Was Uploaded	63
Summary	63
Preview of Other Features	64

Objectives

At the end of this tutorial, you will have accomplished the following:

- Added source files in Dimensions to a Dimensions project
- Compiled the *.c and *.h files in the PRJ_DEV_WIN project for the Payroll example
- Produced a Payroll.exe file as the result of the build
- Checked the result of the build back into Dimensions

Before You Begin

Before you start this tutorial, please be sure that the following is true:

- You have a working installation of Dimensions.
- You have installed the Payroll example, and have access to the PRJ_DEV_WIN project.
- You have set the default item library properly for the Payroll project.
- You have access to the Build Administration features in the Dimensions Administration Console.
- You have a working C compiler (the exercises assume the gcc compiler).

Also, be sure you are familiar with the concepts presented in "[C Quick Start, Part One](#)," namely:

- You know how to add a Dimensions project to the Build Administration console.
- You know how to create a build configuration.
- You know how to create targets.
- You know how to run a build and how to examine the build logs in case something goes wrong.

Exercise 1 Create a New Work Area

For this exercise, you will create a new work area called PRJ_DEV_WIN_workarea. This work area will contain the files from the Payroll example.

To create the new work area:

- 1 Log in to the Dimensions Administration Console.
- 2 Find and click on **Area Definitions**. The Area Definitions page appears.
- 3 Click the New icon and choose Work Area.

- 4 Fill out the fields as follows:

- **ID**

Enter PRJ_DEV_WIN_WORK_AREA.

- **Description**

Enter a description for the build area.

- **Network Node**

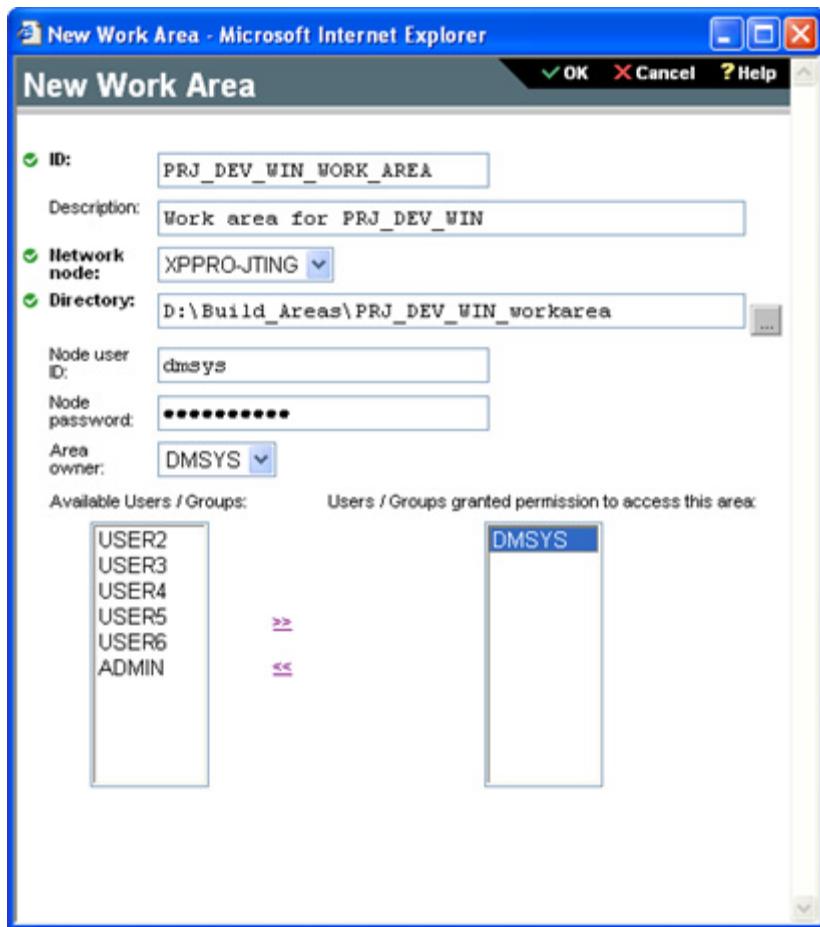
Enter the name of the machine that Dimensions is to use when executing the build.

- **Directory**

Enter the name of the directory for the work area; in this example, we will use "D:\Build_Areas\PRJ_DEV_WIN_workArea".

For the remaining fields, use the user ID and password you used in [C Quick Start, Part One](#). In the examples given here, the user ID will be "dmsys".

When you are done, the dialog box should look like this:



- 5 Click **OK** and verify that the new area has been created.

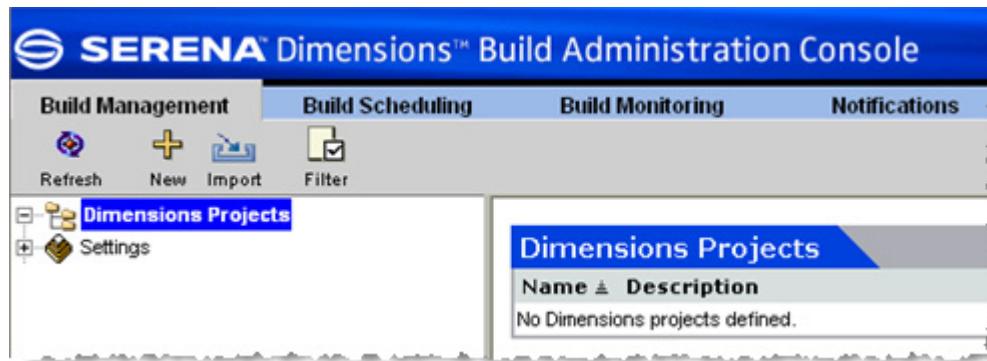
- 6 Click **Home** to return to the main page of the Dimensions Administration Console.

Exercise 2 Add the Payroll Project Build Configuration

For this exercise, you will create a build configuration and, in the process, add the Payroll project to Dimensions.

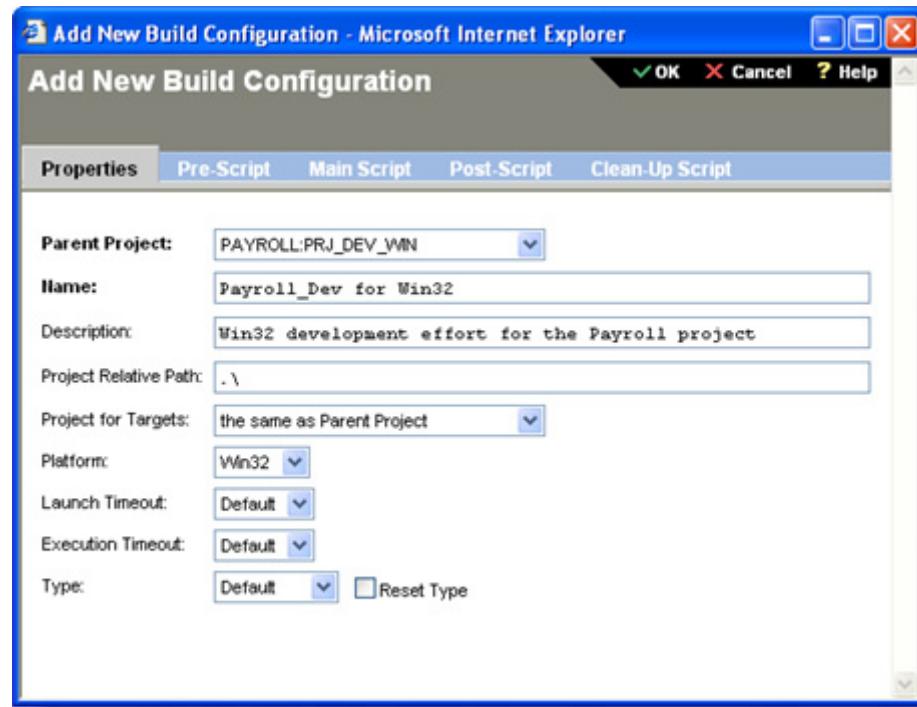
To create the Payroll project build configuration:

- From the Dimensions Administration Console, click **Build administration**. The Dimensions administration page appears.



- Click **New**. The Add New Build Configuration dialog box appears.
- Fill out the fields on the Properties tab as follows:
 - Parent Project**
Select **PRJ_DEV_WIN**.
 - Name**
Enter the name **Payroll_Dev for Win32**.
 - Description**
Enter a description such as **Win32 development effort for the Payroll project**.
 - Project Relative Path**
Enter **.**
 - Project For Targets**
Accept the default choice of **the same as Parent Project**. This field is important as it determines where the target produced by the build will be checked in to Dimensions.
 - Platform**
Accept the default of **Win32**.
- Leave the fields on the other tabs unchanged.

When you are done, the dialog box should look like this:



- 5** Click **OK**.
- 6** Verify that the new build configuration has been added, and that the PRJ_DEV_WIN project appears in the Dimensions Projects tree.
- 7** Select the name of the newly-created build configuration. The right side of the application, known as the content area, displays details for the build areas, build targets, and build options.

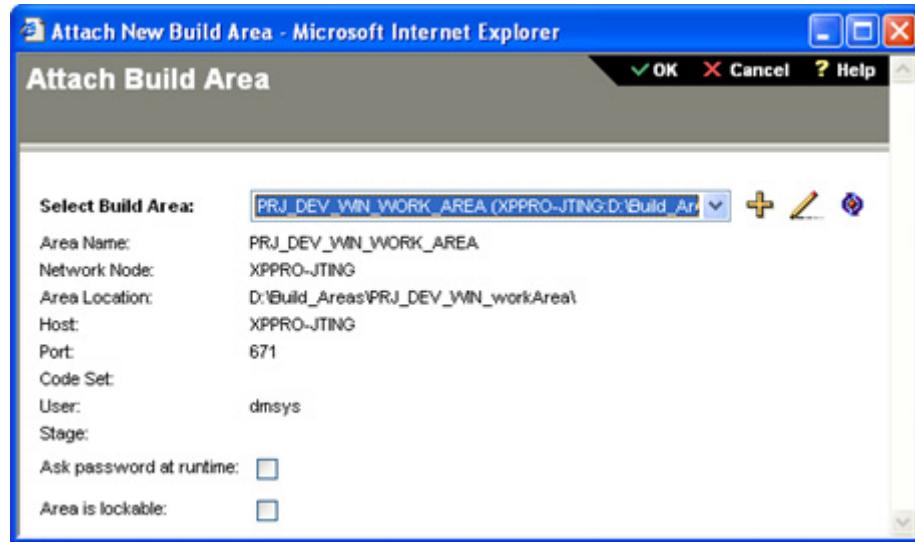
Exercise 3 Attach the Build Area

In this exercise, you will attach the earlier-created build area to the build configuration you created in the last exercise.

To attach the build area:

- 1** Click the New object icon (shaped like a plus sign) next to the Build Areas section title. The Attach New Build Area dialog box appears.
- 2** Select the build area created earlier.
- 3** Accept the default for all other fields.

When you are done, the dialog box should look like this:



- 4 Click OK. Verify that the new build area has been attached to the build configuration.

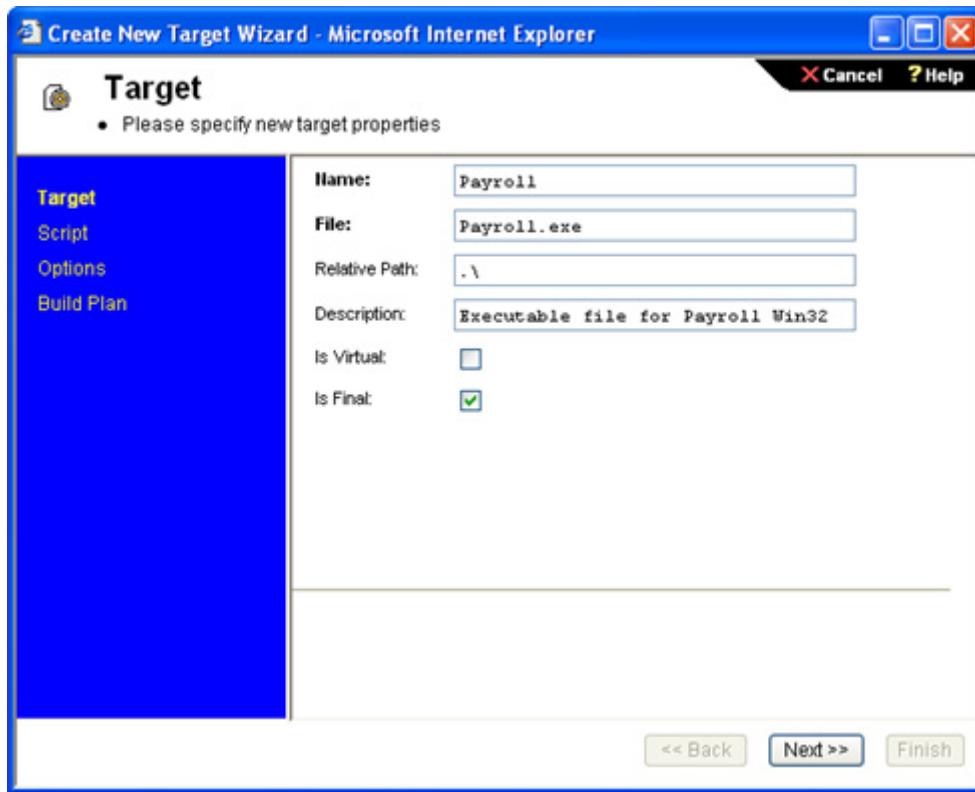
Exercise 4 Define the Build Target

In this exercise, you will define the build target.

To define the build target:

- 1 Click the New object icon (shaped like a plus sign) next to the Build Targets heading. The Create New Target wizard appears.
- 2 Fill out the fields on the Target page as follows:
 - **Name**
Enter **Payroll**.
 - **File**
Enter **Payroll.exe**.
 - **Relative Path**
Enter **\.**
 - **Description**
Enter a description for the build target such as **Executable file for Payroll Win32**.
- 3 Accept the default values for the remaining fields on this page.

- 4 When you are done with the Target page, it should look like this:



- 5 Click **Next**. The Script page appears. Go on to the next exercise.

Exercise 5 Define the Build Script

In this exercise, you will define the main build script for the target. This exercise in particular assumes that you have completed the previous exercise.

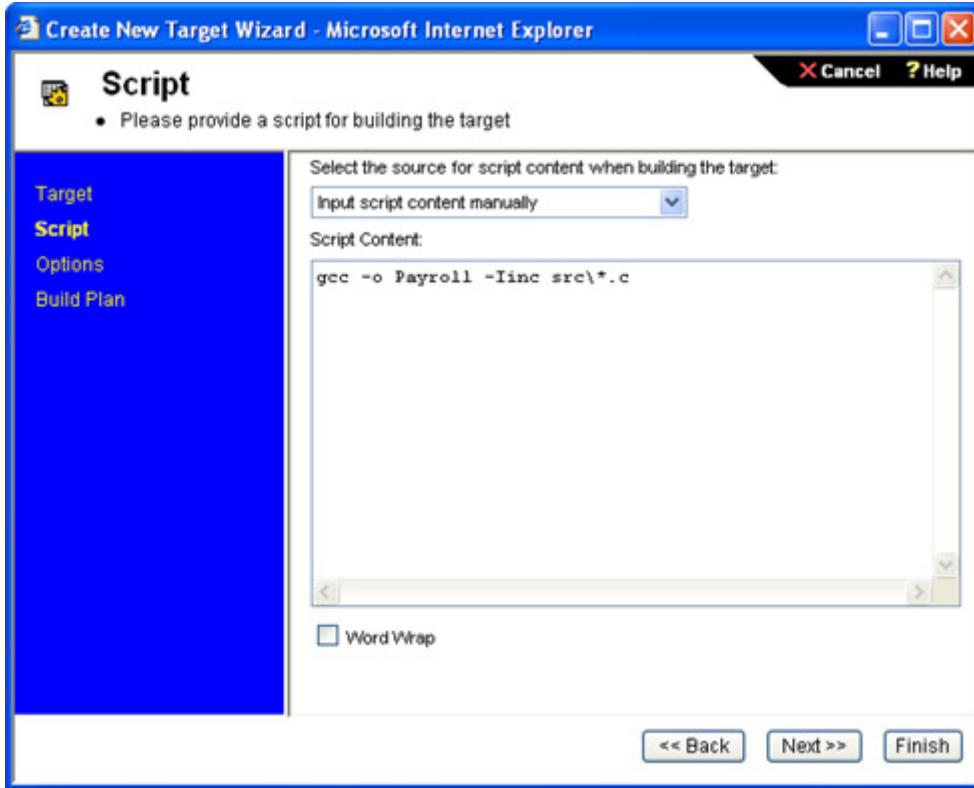
To define the main build script:

- From the Script page of the Create New Target Wizard, select **input script content manually** and enter the following command to compile your C program.

```
gcc -o Payroll -Iinc src\*.c
```

This tells the gcc compiler to produce an executable file named Payroll, to search the inc directory for include files, and to compile all .c files in the src directory.

When you are done, the dialog box should look like this:



- 2 Since there is no need to fill out any of the remaining pages of the wizard, click **Finish**.



NOTE Although it is possible to define sources using this wizard, the tutorial will cover a different way of doing that, in a later exercise.

- 3 Verify that the build configuration and the build target were created.

Exercise 6 Add Files from Dimensions

In this exercise, you will add source and include files that are already in Dimensions to the build configuration.

To add source files from Dimensions:

- 1 Select the name of the build configuration if it is not selected already.

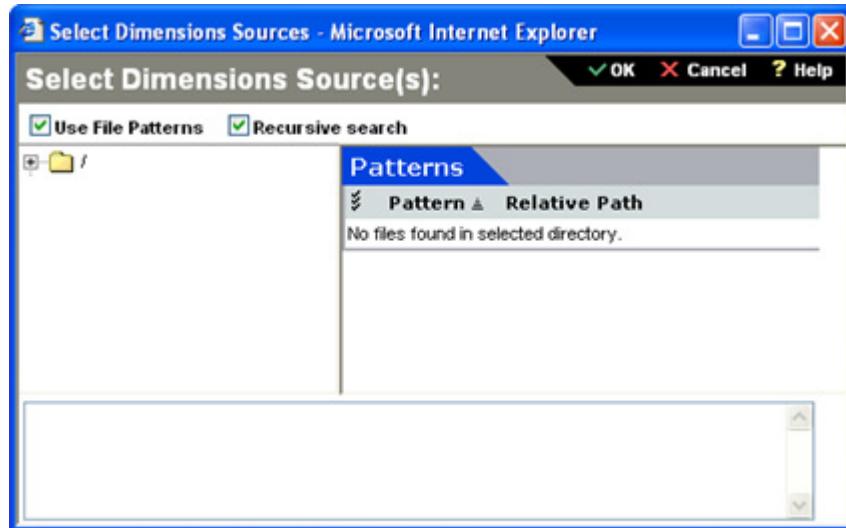
- 2** In the right-hand part of the Dimensions Administration Console, next to the Build Targets heading, click **Sources**:



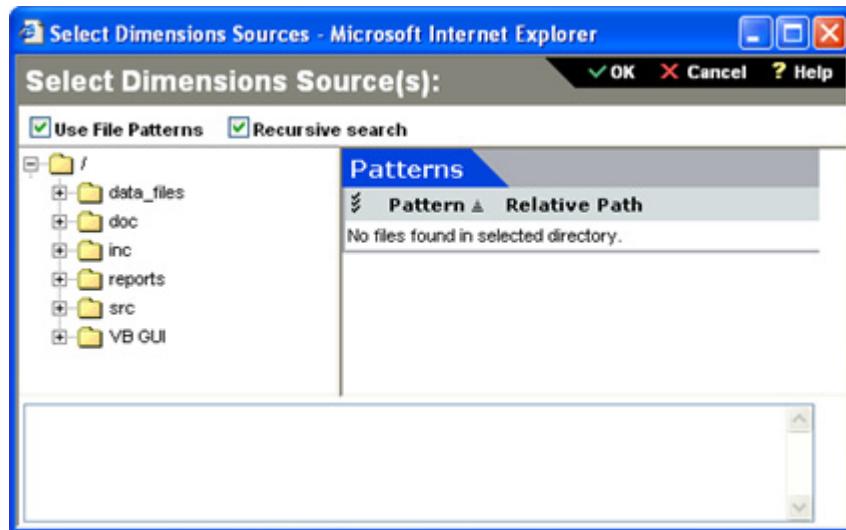
- 3** Click the **Add sources from Dimensions** icon:



The Select Dimensions Sources dialog box appears:



- 4** Expand the directory tree until the files for the Payroll project appear:



- 5 Select the **src** folder. Two entries appear in the Patterns area.
- 6 Check the box for the ***.c** pattern. Notice the wildcard string that appears in the horizontal box at the bottom of the dialog box.
- 7 Click **OK**.
- 8 Verify that the pattern for *** . c** files appears in the list of Sources.

To add include files from Dimensions:

- 1 Repeat Step 1 through Step 7 but select the **inc** folder in Step 5 and check the box for the ***.h** pattern in Step 6.
- 2 Click **OK**.
- 3 Verify that the pattern for *** . h** files appears in the list of Sources.

Exercise 7 Add Existing Sources as Inputs

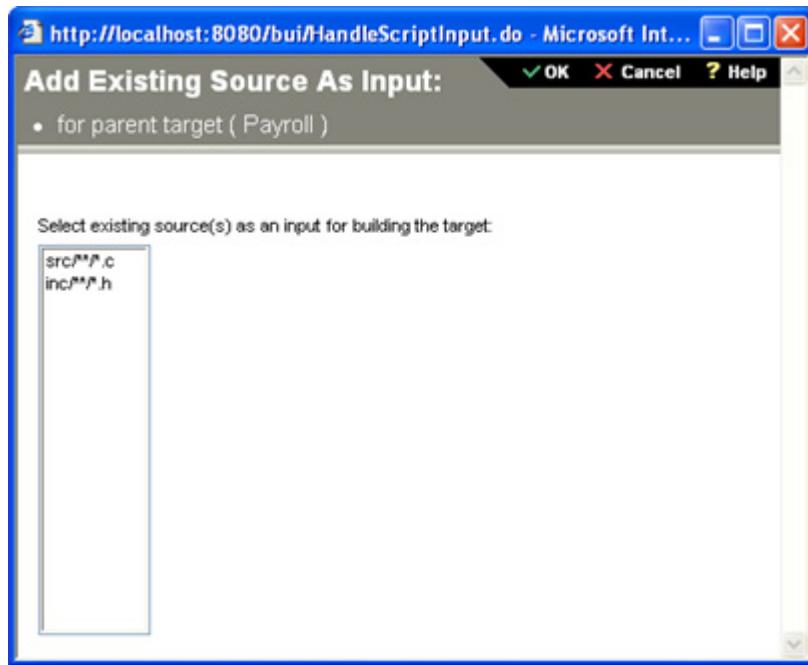
In this exercise, you specify the source files added in the previous exercise as inputs to the build.

To add existing sources as inputs:

- 1 Select the name of the target (in this case, **Payroll**).
- 2 In the right-hand part of the Dimensions Administration Console, click **Inputs**.
- 3 Click the **Add existing source(s) as an input** icon:



The **Add Existing Source as Input** dialog box appears:



- 4 Use Ctrl-Click to select the pattern for ***.c** and the pattern for ***.h**.



NOTE It is best if you use wildcard patterns when you have a lot of sources to add, rather than naming or select all the files individually.

- 5 Click **OK**.
- 6 Verify that the two pattern specifications appear in the list of Inputs.

Exercise 8 Check in the Build Configuration

In this exercise, you check in the build configuration. Remember that until a build configuration is checked in, its changes will not be executed.

To check in the build configuration:

- 1 Click the **Check In** icon:



The Provide check-in comment dialog box appears.

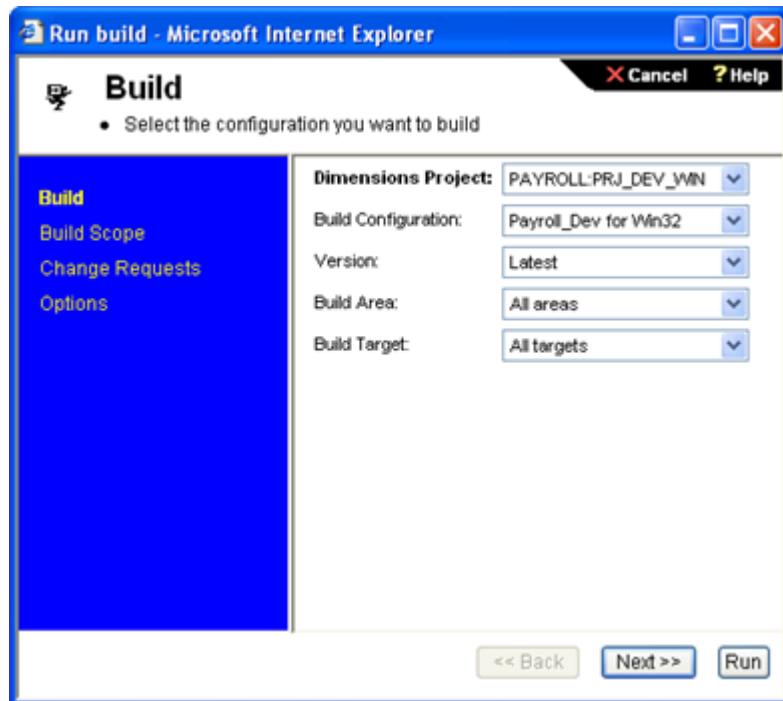
- 2 Enter a comment describing the changes, then click **OK**. A Results dialog box displays a confirmation that the build configuration was checked in.

Exercise 9 Run the Build

In this exercise, you run the build job.

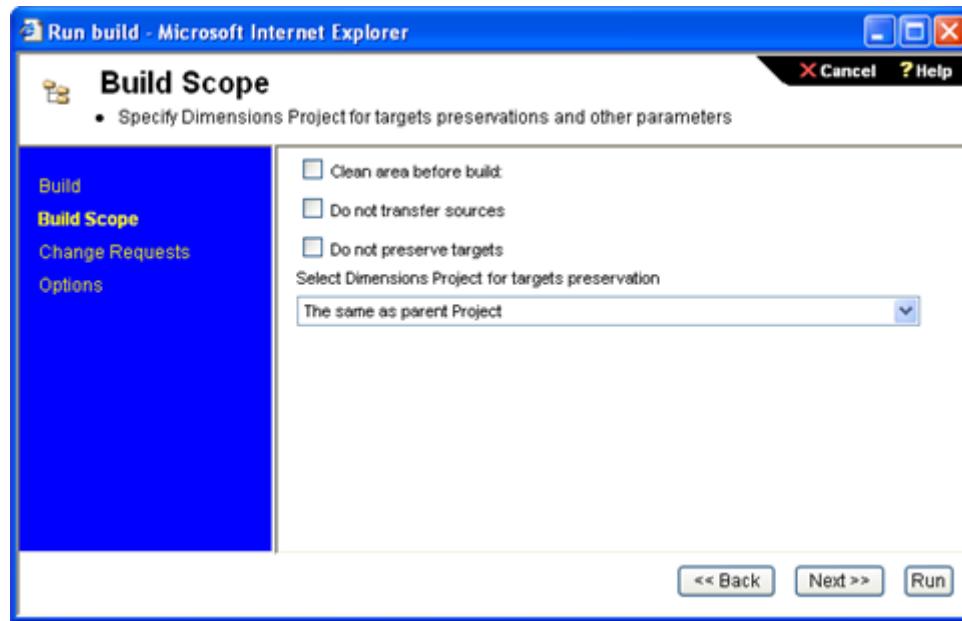
To run the build job:

- 1 Select the name of the build configuration if it is not selected already. The **Run** icon appears.
- 2 Click the Run icon. The Run Build wizard appears.



- 3 If you have defined more than one build area, select the build area defined earlier in this tutorial.
- 4 If you have defined more than one build target, select the build target defined earlier in this tutorial.

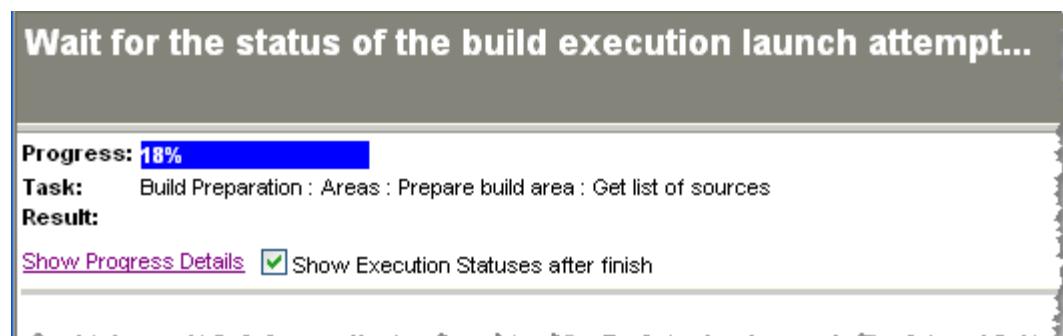
- 5 Click Next. The Build Scope page appears.



The options on this page have the following effects:

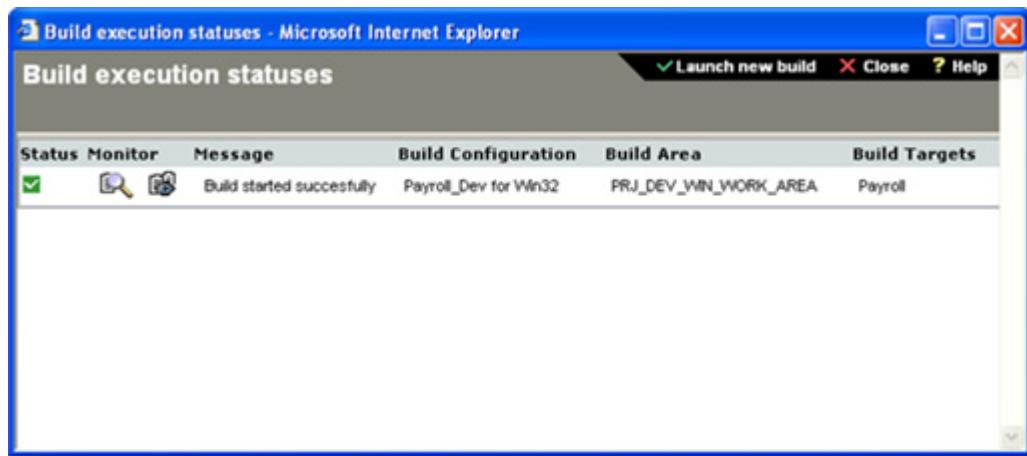
- **Clean area before build** will cause the Clean-Up Script to be executed before any source files are transferred into the build area
- **Do not transfer sources** will cause the build to occur without transferring the sources into the build area
- **Do not preserve targets** will cause the target produced by the build to not be checked in to Dimensions

- 6 Accept the default settings, as shown above, and click **Run**. The **Wait for the status of the build execution launch attempt** dialog box appears:



A progress bar shows the progress of the launch attempt.

- 7 After the launch attempt succeeds, the **Build execution statuses** dialog box appears:



Exercise 10 Verify that Target Was Created

In this exercise you will verify that the Payroll.exe target was created.

To verify that the target was created:

- 1 Open a Command Prompt window.
- 2 Navigate to the build area.
- 3 Verify that the Payroll.exe file was created.
- 4 If you like, run the file. The output will look like this:

```
D:\Build_Areas\PRJ_DEV_WIN_workarea>Payroll
Payroll Initialization for employee completed OK
Payroll Running
No Tax to Pay
No employees added for Win 95
No employees removed for Win 95
Employee Modified for Win 95
Staff Database Updated
Employee deserves a huge bonus
Calculation complete for holiday
Employee deserves a day off!!
Backup completed with no errors
Exiting Payroll Run for
*** Payroll Run COMPLETED for: ***
```

Exercise 11 Verify that Target Was Uploaded

In this exercise you will verify that the target produced by the build job was indeed uploaded to Dimensions.

To verify that the target was uploaded:

Start by examining the output log in the Build Monitor events.

Order	Step	Type
1		Build started
2		Target Build sta
3		Target Build fini
4		Information
5		View output log

If the build target was created successfully, this will be shown in the log results, which will look something like this:

File deployment statuses:
SUCCESS: Target Payroll.exe uploaded.

The best way, of course, is to actually navigate to the project in Dimensions:

The screenshot shows the Dimensions software interface. On the left, there is a navigation tree with the following items: PAYROLL:PRJ_DEV_WIN, Folders and Items (selected), Parts and Items, Work Areas, and Deployment Areas. On the right, there is a table titled "Contents of: PAYROLL:PRJ_DEV_WIN\Payroll.exe". The table has columns: Name, Revision, and Status. There is one row: Payroll.exe, win_xp#3, BUILT.

Name	Revision	Status
Payroll.exe	win_xp#3	BUILT

Remember, the setting of **Do not preserve targets** that was discussed in "[Run the Build on page 60](#)" can cause the targets to not be uploaded to Dimensions. If upload rules have not been set properly, that too can prevent targets from being uploaded.

Summary

In this Quick Start you accomplished the following:

- Repeated the process of creating a build area
- Repeated the process of creating a build configuration and adding projects to Dimensions
- Repeated the process of attaching a build area to a build configuration
- Learned how to add source files that are in Dimensions
- Learned how to use wildcard patterns to add multiple source files
- Learned how the Build Scope settings affect whether or not build targets are uploaded to Dimensions

Preview of Other Features

Dimensions contains many features and this chapter does not cover them all. In subsequent chapters, you will learn about many other features, such as:

- Build Job Scheduling
- Build Job Monitoring
- Notifications
- Importing and exporting build configurations
- Importing Openmake TGT files
- Importing Ant buildfiles

Chapter 3

Assembler MVS Quick Start



NOTE You can only perform this tutorial with Dimensions CM version 10.1.1.

This MVS quick start tutorial shows you how to use Dimensions Build to set up a closed-loop build of a simple Assembler program called Disassembler. In a closed-loop build the targets that you create are captured and check in by Dimensions CM.

What You Will Learn

This tutorial shows you how to perform the following build-related tasks:

- Create a Dimensions project.
- Create mainframe items in the project.
- Add a build configuration to the Dimensions project.
- Create a build area and add it to the build configuration.
- Create build tools and build option groups.
- Create transition rule templates and an application rule template.
- Add build targets to the build configuration.
- Run and monitor the build job.
- View derived items.
- View the bill of materials.

In this tutorial you will use the following Dimensions CM components:

- Dimensions for z/OS ISPF client.
- Dimensions Build (part of the Administration Console).
- Desktop client.

Before You Begin

Pre-Requisites

Before you start the tutorial check the following pre-requisites:

- You have installed, configured, licensed, and started a Dimensions 10.1.1 server on the host machine. For details see the *Installation Guide*. As part of the installation you have installed the *Payroll* or *Qlarius* sample process models.



NOTE The examples and screen shots in the tutorial use the *Payroll* product.

- You have access to a user ID with administrator privileges (such as "dmsys") that is a member of an Administrator group.
- You have installed and licensed Dimensions for z/OS on the machine that will host your build area. For details see the installation chapter in the *Dimensions for z/OS User's and Administrator's Guide*.
- You have setup the following Dimensions network nodes:
 - A physical network node for the z/OS machine where the Dimensions Listener resides.
 - A logical network node and connection for the MVS file system.For details about setting up mainframe nodes see the installation chapter in the *Dimensions for z/OS User's and Administrator's Guide*.
- You have verified that the variable DM_TEMPLATE_CATALOG<n> in the Dimensions for z/OS configuration file specifies the data set where build templates are located. If you do not change DM_TEMPLATE_CATALOG<n> the default data set will be used (\$DM_ROOT/templates). For more information about using DM_TEMPLATE_CATALOG<n> see the *Developer's Reference*.

Data Set Naming Convention

This tutorial uses the following data set naming convention:

<base>. <ILQ>. <LLQ>

This convention may have the following format:

<HLQ1.HLQ2>. <ILQ1.ILQ2>. <single LLQ>

where:

- <HLQ> is a high level qualifier.
- <ILQ> is an intermediate level qualifier.
- <LLQ> is a low level qualifier.

For example:

MDH.DIM1010.DISS.SOURCES.FOO(C)

Member Naming Conventions

The members in this tutorial use the following naming convention:

- ASM source code:

ASM(xxxAxxxx)

For example, ASM(MDOARBLD).

- Object modules:

OBJ(xxxAxxxx)

For example, OBJ(MDOARBLD).

- SYSLIN linkers:

SYSLIN(xxxLxxxx)

For example, SYSLIN(MDOLRBLD).

- LOAD modules:

LOAD(xxxLxxxx)

For example, LOAD(MDOLRBLD).

About Build Templates

This tutorial uses templates to build the intermediate and final targets. A template is a customizable text file containing variables and control words. In Dimensions, templates are used to execute builds, execute general commands via remote job execution, control deployments, and construct the body of e-mail messages used for notifications. For details about templates and the templating language see the *Developer's Reference*.

The templates that you will use in this tutorial are packaged with Dimensions for z/OS and are located in the default TEMPLATE data set that was created with your Dimensions for z/OS instance:

- MDHBASC0: used to invoke the assembler compiler.
- MDHBLNK1: used to invoke the linker compiler.

Exercise 1 Allocate the Build Area Data Sets



IMPORTANT! Any build areas that you set up for managed build areas should be allocated under a security protected ID that normal Dimensions users do not have write access to. This precaution is necessary to protect the integrity of the data contained in the build areas.

To allocate build area data sets:

As an administrator allocate the following data sets:



NOTE Not all the data sets listed below are required for Disassembler, but are needed for possible build areas for other applications.

Name	DSORG	RECFM	LRECL	BLKSZ
<HLQ>.<ILQ>.GEN.ASM	PO-E	FB	80	32720
<HLQ>.<ILQ>.GEN.ASMC	PO-E	FB	80	32720
<HLQ>.<ILQ>.GEN.ASMLIST	PO-E	FBA	121	27951
<HLQ>.<ILQ>.GEN.ASMMAC	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.ASMTERM	PO-E	FBA	121	12100
<HLQ>.<ILQ>.GEN.C	PO-E	FB	80	32720
<HLQ>.<ILQ>.GEN.CBLLIST	PO-E	FBA	133	27930
<HLQ>.<ILQ>.GEN.CBLTERM	PO-E	FBA	80	3120
<HLQ>.<ILQ>.GEN.CLIST	PO-E	FB	80	32720
<HLQ>.<ILQ>.GEN.CNTL	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.COBC	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.COBOL	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.CONTROL	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.COPY	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.CPP	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.CPPSQL	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.DATA	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.DEPENDCY	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.H	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.IMPORT	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.INCLUDE	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.JCLLIB	PO-E	VB	1028	27998
<HLQ>.<ILQ>.GEN.LNK1LIST	PO-E	FBA	121	1210
<HLQ>.<ILQ>.GEN.LNK2LIST	PO-E	FBA	121	1210
<HLQ>.<ILQ>.GEN.LNKLIB	PO-E	U	0	32760

Name	DSORG	RECFM	LRECL	BLKSZ
<HLQ>.<ILQ>.GEN.LNKTERM	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.LOAD	PO-E	U	0	32760
<HLQ>.<ILQ>.GEN.LOADMOD	PO-E	U	0	32760
<HLQ>.<ILQ>.GEN.MACLIB	PO-E	FB	80	32720
<HLQ>.<ILQ>.GEN.MDHFLAT	PO	FB	80	3120
<HLQ>.<ILQ>.GEN.MSGS	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.OBJ	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.PANELS	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.PLI	PO-E	FB	80	32720
<HLQ>.<ILQ>.GEN.PLILIST	PO-E	VBA	125	23476
<HLQ>.<ILQ>.GEN.SKELS	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.SYSLIN	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.TABLES	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.TEMPLATE	PO-E	FB	80	3120
<HLQ>.<ILQ>.GEN.TGT	PO	VB	32750	32760

Exercise 2 Copy the Disassembler Sources to the Build Area Data Sets

In this exercise you will copy the Disassembler sources from the Dimensions for z/OS distribution to the build area data sets that you created in the previous step.

To copy the Disassembler sources to the build area data sets:

- 1 Open an ISPF session.
- 2 Use the utilities function to copy the contents of MDH.V1011.SUPP.DISASS.ASM to '<HLQ>.<ILQ>.GEN.ASM'.
Verify that the following files were copied:
 - MDOARBLD
 - MDOARCHK
 - MDOARDLM
 - MDOARSRC
- 3 Copy the contents of MDH.V1011.SUPP.DISASS.SYSLIN to '<HLQ>.<ILQ>.GEN.SYSLIN'.

- 4 Verify that the following files were copied:

- MDOLRBLD
- MDOLRCHK
- MDOLRSRC

Exercise 3 Create a Dimensions Project and Directories

In this exercise will you use the Command Entry panel in the Dimensions for z/OS ISPF client to create a project and directories for Disassembler in the *Payroll* or *Qlarius* products.

To create a Dimensions project and directories:

- 1 Log in to the Dimensions for z/OS ISPF client using the log in parameters provided by your administrator.
- 2 From the Commands menu select Command Entry. The Command Entry panel is displayed.
- 3 In the Dimensions command field enter the following command to create a new project called DISASSEMBLER:

```
DWS "PAYROLL:DISASSEMBLER" /DESCRIPTION="Project for Disassembler"
```

or

```
DWS "QLARIUS:DISASSEMBLER" /DESCRIPTION="Project for Disassembler"
```

For example:



Enter a Dimensions command in the field below:
==> **DWS "PAYROLL:DISASSEMBLER" /DESCRIPTION="Project for Disassembler"**

- 4 Press Enter. The message 'Executing' is displayed in the top right corner. After the command has finished executing, the Dimensions Command Log panel is displayed.
- 5 To close the Dimensions Command Log panel and return to the Command Entry panel press <END>.
- 6 In the Dimensions command field enter the following command to create a new directory called ASM in the project DISASSEMBLER:

```
CWSD ASM /WORKSET=PAYROLL:DISASSEMBLER
```

or

```
CWSD ASM /WORKSET=QLARIUS:DISASSEMBLER
```



NOTE The ASM directory will be used for ASM source files.

- 7 Press Enter. The message 'Executing' is displayed in the top right corner. After the command has finished, executing the Dimensions Command Log panel is displayed.
- 8 To close the Dimensions Command Log panel and return to the Command Entry panel press <END>.
- 9 Repeat steps 3–8 to create the following directories:

Directory Name	Purpose
LOAD	Directory for the final build targets.
OBJ	Directory for intermediate OBJ targets.
SYSLIN	Directory for SYSLIN source files.

- 10 To return to the ISPF client main panel press <END>.



NOTE You can also use the desktop and web clients to add a Dimensions project and directories.

Exercise 4 Set the Project and the Project Root

In this exercise you will use the ISPF client to set the current project to PAYROLL:DISASSEMBLER or QLARIUS:DISASSEMBLER and set the project root to your GEN work area.

To set the project and the project root:

- 1 From the Commands menu select Set Current Project. The Set Current Project panel is displayed.
- 2 If your current project is not PAYROLL or QLARIUS, in the **Product** field enter '/' and press Enter. The Product Selection pop-up panel is displayed. Enter 's' next to PAYROLL or QLARIUS and press Enter. The Product Selection pop-up panel closes.
- 3 In the **Project** field enter '/' and press Enter. The Project Selection pop-up panel is displayed. Enter 's' next to DISASSEMBLER and press Enter. The Project Selection pop-up panel closes.
- 4 In the **Project Root** field enter '/' and press Enter. The Set Project Root pop-up panel is displayed.
- 5 In the **Node Name** field enter '/' and press Enter. The Network Node pop-up panel is displayed. Enter 's' next to the logical mainframe network node where you will be building DISASSEMBLER and press Enter. The Network Node pop-up panel closes.
- 6 In the **Dataset** field enter the full data set name of the project root, for example, '<HLQ>. <ILQ>. GEN'. Press Enter. The Set Project Root pop-up panel closes.
- 7 To make DISASSEMBLER the default project, enter '/' in the **Make default Project** field.

- 8 Press Enter. The Set Current Project panel closes. The ISPF client main panel refreshes and updates the name of the current project. Your ISPF main panel should look similar to this (to refresh the ISPF client press PF5):

The screenshot shows the ISPF client main panel titled "Serena Dimensions". The "Command ==>" field is empty. Below it, the "Project path:" field shows the connection and project details: "Connection:dm-development:671[intermediate@dim10ga]", "Project: PAYROLL: DISASSEMBLER", "Project root:mvs-01:mdhdev.bld.gen", and "Project path:". A table below lists file types: ASM, LOAD, OBJ, and SYSLIN, each preceded by a plus sign (+). The table has columns for "Filename", "Revision", and "Status".

Filename	Revision	Status
+ ASM		
+ LOAD		
+ OBJ		
+ SYSLIN		

Exercise 5 Create the Items in Dimensions

In this exercise you will use the ISPF client to create the ASM and SYSLIN source items in Dimensions CM.

To create the ASM source items:

- 1 In the ISPF client main panel enter 's' next to ASM. Press Enter. The ISPF client refreshes and displays the contents of the ASM directory (which is currently empty). The Project Path field displays the current project path.
- 2 From the Item menu select Create. The Create Item panel is displayed.
- 3 To keep the source file in the distribution data set, in the **Keep copy in user area** field enter '/'.
- 4 In the **Dataset name** field enter the name of the data set containing the first ASM source file that you are going to add to Dimensions:

<node_name>::<HLQ>.<ILQ>.GEN.ASM(MDOARBLD)

- 5 To automatically populate the fields in the panel with default values, press F5.
- 6 In the **Comments** field enter *ASM source*.
- 7 In the Filename field add the extension .ASM after the filename.

Your Create Item panel should look similar to this:

Create Item

Command ==> _____

Source Dataset _____ Enter "/" to select option
Project . . . _____ / Keep copy in user area
Group . . . _____
Type . . . _____
Member . . . _____
OR
Dataset name . . <node name>::<HLQ>.<ILQ>.GEN.ASM(MDOARBLD)

Revision description . . Initial Revision

Owning design part . PAYROLL:PAYROLL.A;1

Item format . . . GENBINARY _____ Enter "/" to select

Item specification . PAYROLL:MDOARBLD GEN.A-DAT;1 _____ Enter "/" to select

Project Filename

Directory . . . ASM

Filename . . . MDOARBLD.ASM

Library filename . . mdoarbl -01.gen

Request(s) . . . _____ Enter "/" to select

Comment . ASM source

Options . _____

- 8 Press Enter. After the item is successfully created in Dimensions the following message is displayed in the top right corner of the ISPF client main panel:

Item Create Successful
- 9 Repeat steps 3–8 to add the following ASM sources:
 - <node name>::<HLQ>.<ILQ>.GEN.ASM(MDOARCHK)
 - <node name>::<HLQ>.<ILQ>.GEN.ASM(MDOARDLM)
 - <node name>::<HLQ>.<ILQ>.GEN.ASM(MDOARSRC)
- 10 To exit the Create Item panel press PF3.

- 11 To refresh the ISPF client press PF5. The ASM directory now shows the new items.

Project path: ASM			
	Filename	Revision	Status
-	MDOARBBLD.ASM	1	UNDER WORK
-	MDOARCHK.ASM	1	UNDER WORK
-	MDOARDLM.ASM	1	UNDER WORK
-	MDOARSRC.ASM	1	UNDER WORK

- 12 Press PF3 to return to the top level directory in the DISASSEMBLER project.

To create the SYSLIN source items:

- 1 In the ISPF client main panel enter 's' next to SYSLIN. Press Enter. The ISPF client refreshes and displays the contents of the directory (which is currently empty). The Project Path field displays the current project path.
- 2 From the Item menu select Create. The Create Item panel is displayed.
- 3 To keep the source file in the distribution data set, in the **Keep copy in user area** field enter '/'.
- 4 In the **Dataset name** field enter the name of the data set containing the first SYSLIN source file that you are going to add to Dimensions:
`<node name>::<HLQ>.<ILQ>.GEN.SYSLIN(MDOLRBLD)`
- 5 To automatically populate the fields in the panel with default values, press F5.
- 6 In the Filename field add the extension .SYSLIN after the filename.
- 7 In the **Comments** field enter *SYSLIN source*.
- 8 Press Enter. After the item is successfully created the following message is displayed in the top right corner of the ISPF client main panel:
`Item Create Successful`
- 9 Repeat steps 3–8 to add the following SYSLIN sources:
 - `<node name>::<HLQ>.<ILQ>.GEN.SYSLIN(MDOLRCHK)`
 - `<node name>::<HLQ>.<ILQ>.GEN.SYSLIN(MDOLSRC)`
- 10 To exit the Create Item panel press PF3.

- 11** To refresh the ISPF client press PF5. The SYSLIN directory now shows the new items.

Project path: SYSLIN			
	Filename	Revision	Status
-	MDOLRBLD.SYSLIN	1	UNDER WORK
-	MDOLRCHK.SYSLIN	1	UNDER WORK
-	MDOLRSRC.SYSLIN	1	UNDER WORK

- 12** Press PF3 to return to the top level directory in the DISASSEMBLER project.

Exercise 6 Add a Build Configuration to the Dimensions Project

In this exercise you will use Dimensions Build to add a build configuration to the Disassembler project that you created earlier. A build configuration captures the information about what will be built, including the targets, build areas, options, scripts, and dependencies for a specific platform or environment.

To add a build configuration to the Dimensions project:

- 1 Log in to the Administration Console using the URL and log in parameters provided by your administrator.
- 2 In the Administration Console, in the Distributed Development cluster, click Build Administration.



The Dimensions Build administration console appears and has the following tabs:

- Build Management
- Build Scheduling
- Build Monitoring
- Notifications

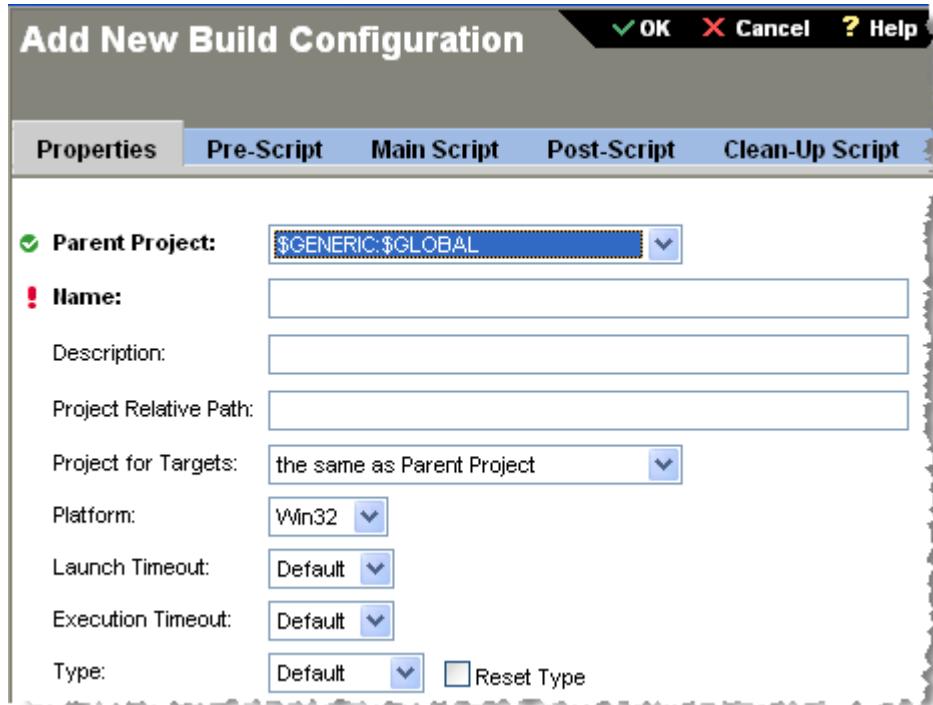


In this tutorial you will mainly be using the Build Management tab. By default the Build Management tab is on top.

- 3 In the menu area click New.



The Add New Build Configuration dialog box appears. There are five tabs on this dialog box. In this tutorial you will only be using the Properties tab, which by default is on top.



- 4 From the **Parent Project** list select PAYROLL:DISASSEMBLER.
- 5 For **Name** type *Disassembler*.
- 6 For **Description** type *Disassembler build configuration*.
- 7 Skip the **Project Relative Path** field.
- 8 From the **Projects for Targets** list select *the same as Parent Project*.
- 9 From the **Platform** list select *MVS*.
- 10 From the **Launch Timeout** list select Default. Launch Timeout specifies the interval in seconds before Dimensions Build abandons a build job whose launch attempt has stalled.
- 11 From the **Execution Timeout** list select Default. Execution Timeout specifies the interval in seconds before Dimensions Build abandons a build job that has stalled due to problems such as script errors or a slow network.
- 12 From the **Type** list select Default. A build configuration type specifies options that are unique to a particular build configuration and are required to run a build. For more details see [page 152](#).

- 13** Click OK. The build configuration *Disassembler* is added to the Disassembler build project. Note that the icon next to the build configuration has a red check mark indicating that the new build configuration is checked out.



Exercise 7 Add a Build Area to the Build Configuration

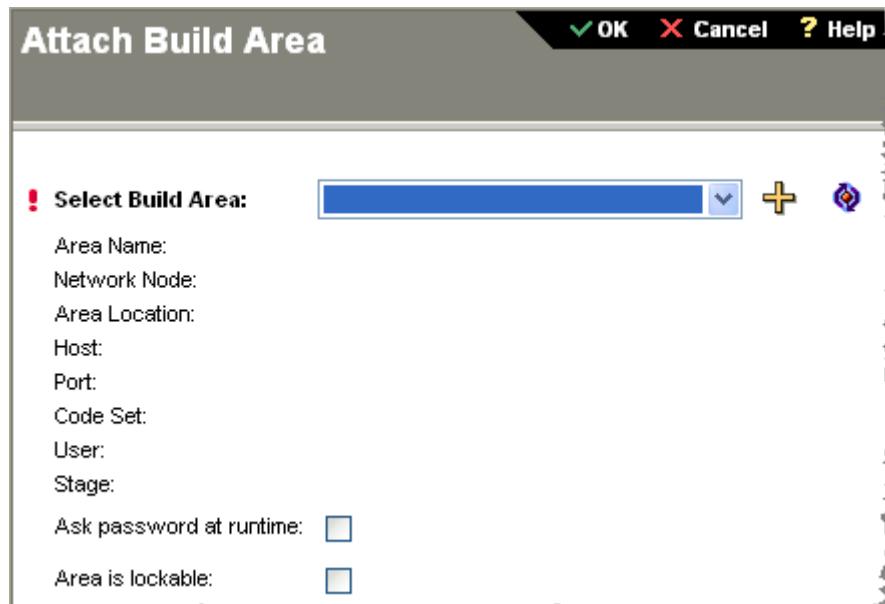
In this exercise you will use Dimensions Build to add a work area to the build configuration. The work area is on the mainframe node where Disassembler will be built. In Dimensions Build work areas are referred to as build areas.

To add a build area to the build configuration:

- 1 In the navigation area of the Build Management tab select the build configuration Disassembler.
- 2 In the content area, in the Build Areas section, click Add New Object.



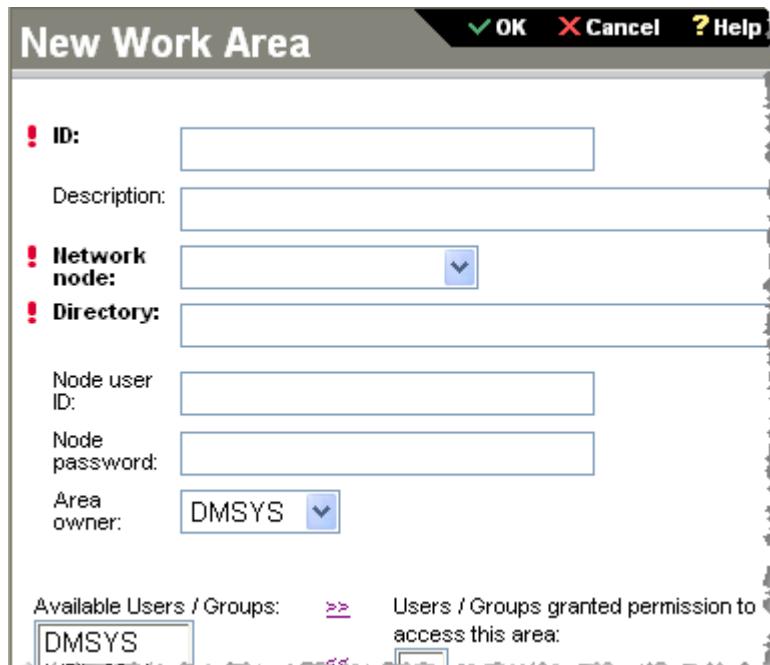
The Attach Build Area dialog box appears.



- 3 Click Add New Dimensions Area.



The New Work Area dialog box appears.



- 4 For **ID** type the following identifier for the build area: *DISASSEMBLER-GEN*.
- 5 For **Description** type the following description for the build area: *GEN build area for Disassembler*.
- 6 From the **Network Node** list select the logical node name of your MVS machine. This is the logical network node specified in the pre-requisites for this tutorial on [page 66](#).
- 7 For **Directory** type the name of the GEN data set that you created in a previous exercise: '*<HLQ>.ILQ.GEN*'. This is the data set that Dimensions will use as the base data set for the build.
- 8 For **Node User ID** type the mainframe user ID that has access to the work area on the mainframe node. This user ID should have full read/write access to the work area.
- 9 For **Node Password** type the password of the user ID that you specified in the previous step.
- 10 From the **Area Owner** list select the Dimensions user ID that will own the work area after it is created and has permission to edit it, for example, *DMSYS*.
- 11 (Optional) In the **Available Users/Groups** list select one or more Dimensions user IDs that will be able to modify this work area, and click the **>>** button. The user IDs are added to the list of **Users/Groups granted permission to access this area**.
- 12 To create the work area click OK. The New Work Area dialog box closes.

- 13 In the Attach Build Area dialog box, from the **Select Build Area** list select DISASSEMBLER-GEN. The fields in the Attach Build Area dialog box are populated with the details of the new work area.
- 14 To attach the build area to the build configuration click OK. The build area is added to the build configuration Disassembler.



Exercise 8 Create Build Tools

In this exercise you will use Dimensions Build to create build tools. Build tools enable you to enforce standards across all builds by managing the options supported by the build tools used in build scripts. For more information see [Chapter 7, "Managing Build Settings" on page 151](#). After you have created the build tools you will add them to build option groups that you will create in the next exercise.



NOTE Creating a build tool is a one time exercise that you do not have to repeat. After you have created a build tool you can add it to any build targets, build option groups, or transition rule templates that you subsequently create.

In this exercise you will add the following build tools:

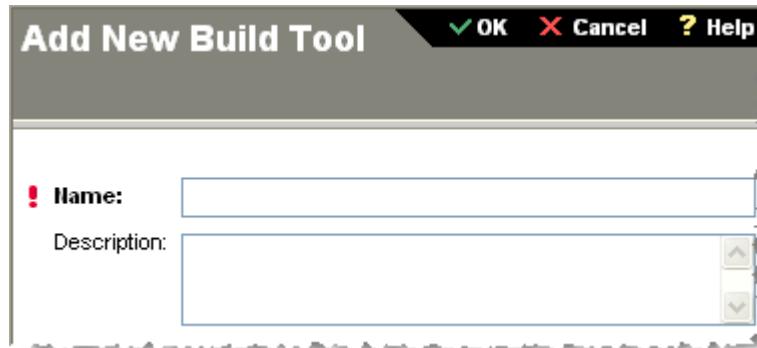
- IBM ASM Compiler: used to assemble OBJ modules.
- IBM Link Tool: used to link LOAD modules.

To create the build tool IBM ASM Compiler:

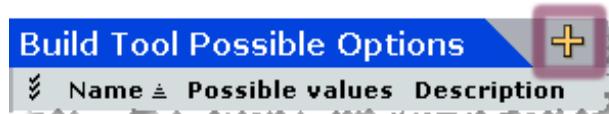
- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the content area click New Object.



The Add New Build Tool dialog box appears.



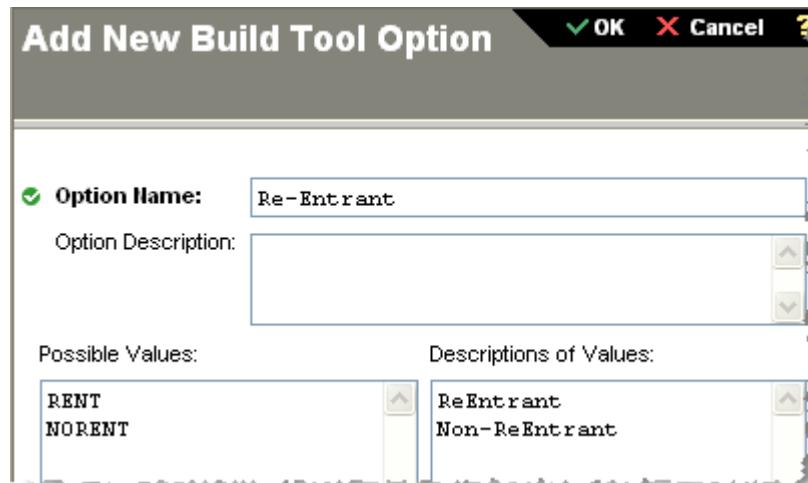
- 3 For **Name** type *IBM ASM Compiler*.
- 4 For **Description** type *High level OBJ assembler*.
- 5 Click OK. IBM ASM Compiler is added to the list of build tools in the navigation and contents areas.
- 6 To add the option Re-Entrant to the build tool IBM ASM Compiler do the following:
 - a In the navigation area select IBM ASM Compiler.
 - b On the Build Tool Possible Options tab in the content area click New Object.



The Add New Build Option dialog box appears.

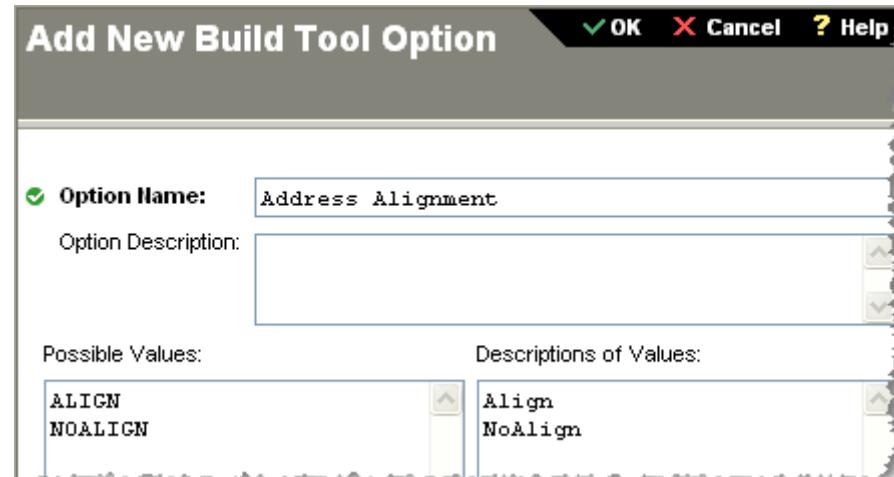
- c For **Option Name** type *Re-Entrant*.
- d Click the **Possible Values** field and type *RENT*. Press the Return key and type *NORENT*.
- e Click the **Description of Values** field and type *ReEntrant*. Press the Return key and type *Non-ReEntrant*.

Your Add Build Tool Option dialog box should look similar to this:



- f** Click OK.
- 7** To add the option Address Alignment to the build tool IBM ASM Compiler do the following:
- In the navigation area select IBM ASM Compiler.
 - On the Build Tool Possible Options tab in the content area click New Object. The Add New Build Tool Option dialog box appears.
 - For **Option Name** type *Address Alignment*.
 - Click the **Possible Values** field and type *ALIGN*. Press the Return key and type *NOALIGN*.
 - Click the **Description of Values** field and type *Align*. Press the Return key and type *NoAlign*.

Your Add Build Tool Option dialog box should look similar to this:



- f Click OK. Your content area for IBM ASM Compiler should look similar to this:

Name	Possible values	Description
Address Alignment	ALIGN NOALIGN	Align No-Align
Re-Entrant	RENT NORENT	Reentrant Non-Reentrant

To create the build tool IBM Link Tool:

- 1 In the navigation area click Build Tools.
- 2 In the content area click New Object.



The Add New Build Tool dialog box appears.

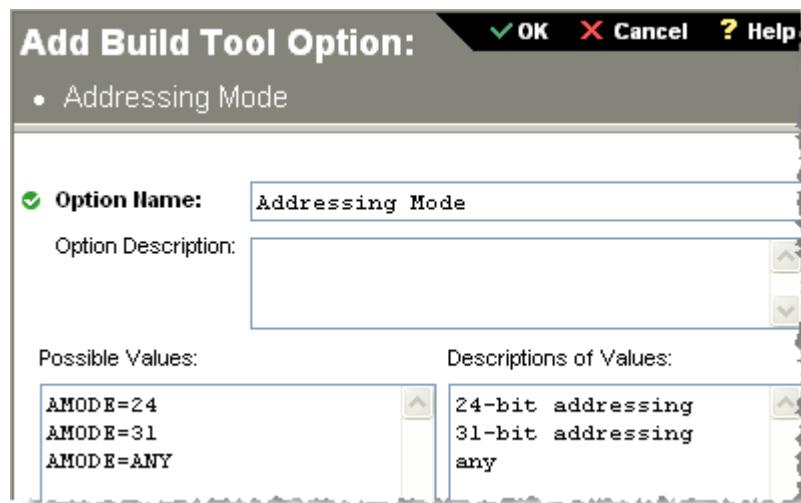
- 3 For **Name** type *IBM Link Tool*.
- 4 For **Description** type *Link LOAD module*.
- 5 Click OK. IBM Link Tool is added to the list of build tools in the navigation and contents areas.
- 6 To add the option Addressing Mode to the build tool IBM Link Tool do the following:
 - a In the navigation area select IBM Link Tool.
 - b On the Build Tool Possible Options tab in the content area click New Object.

Name	Possible values	Description
------	-----------------	-------------

The Add New Build Tool Option dialog box appears.

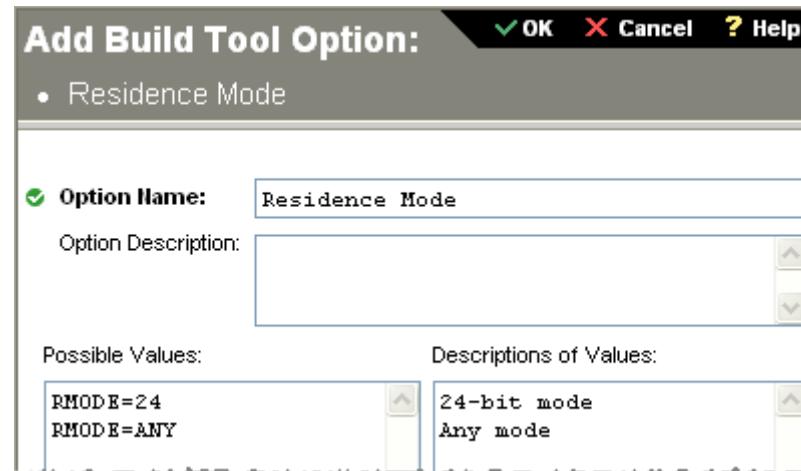
- c For **Option Name** type *Addressing Mode*.
- d Click the **Possible Values** field and type *AMODE=24*. Press the Return key and type *AMODE=31*. Press the Return key and type *AMODE=ANY*.
- e Click the **Description of Values** field and type *24-bit addressing*. Press the Return key and type *32-bit addressing*. Press the Return key and type *any*.

Your Add Build Tool Option dialog box should look similar to this:



- f** Click OK.
- 7** To add the option Residence Mode to the build tool IBM Link Tool do the following:
- In the navigation area select IBM Link Tool.
 - On the Build Tool Possible Options tab in the content area click New Object. The Add New Build Tool Option dialog box appears.
 - For **Option Name** type *Residence Mode*.
 - Click the **Possible Values** field and type *RMODE=24*. Press the Return key and type *RMODE=ANY*.
 - Click the **Description of Values** field and type *24-bit mode*. Press the Return key and type *any mode*.

Your Add Build Tool Option dialog box should look similar to this:



- f Click OK. Your content area for IBM Link Tool should look similar to this:

Build Tool Possible Options		
Name	Possible values	Description
	AMODE=24	24-bit addressing
<input type="checkbox"/> Addressing Mode	AMODE=31	31-bit addressing
	AMODE=ANY	any
	RMODE=24	24-bit mode
<input type="checkbox"/> Residence Mode	RMODE=ANY	Any mode

Exercise 9 Create Build Option Groups

In this exercise you will use Dimensions Build to create build option groups and add the build tools that you created in the previous exercise. Build option groups are logical collections of build tool options that enable you to add collections of build options to a target instead of adding each option separately. For more information see [Chapter 7, "Managing Build Settings" on page 151](#). After you have created the build option groups you will add them to the transition rule templates that you will create in the next exercise.



NOTE Creating a build option group is a one time exercise that you do not have to repeat. After you have created a build option group you can add it to any build targets and transition rule templates that you subsequently create.

In this exercise you will create the following build option groups

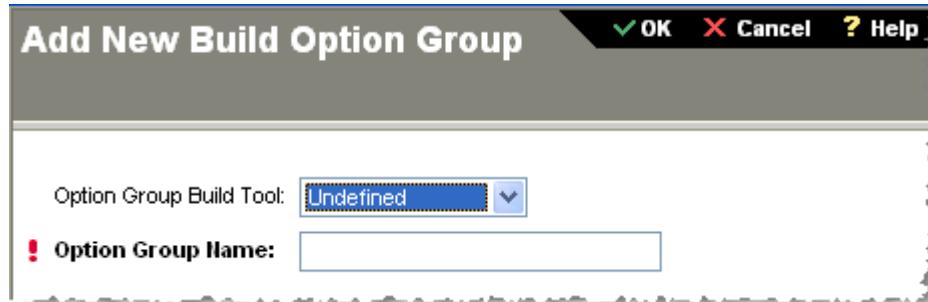
- ASMPARM: includes the option RENT from the build tool IBM ASM Compiler. See below.
- LPARM: includes the options AMODE and RMODE from the build tool IBM Link Tool. See the procedure on [page 86](#).

To create the build option group ASMPARM:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the content or menu areas click New Object.

Build Option Groups	
<input style="background-color: #e0e0e0; color: black; font-weight: bold; font-size: 1em; padding: 2px 5px; border: 1px solid #ccc; border-radius: 5px; width: 15px; height: 15px; vertical-align: middle;" type="button" value="+"/>	
Name ▲ Build Tool	

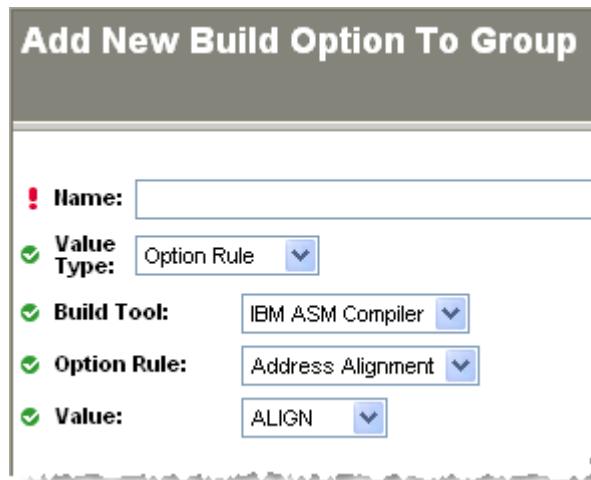
The Add New Build Option Group dialog box appears.



- 3 From the **Option Group Build Tool** list select IBM ASM Compiler.
- 4 For **Option Group Name** type *ASMPARM*.
- 5 Click OK. The new option group ASMPARM is added to the list of build option groups in the navigation and content area.
- 6 To add the RENT option from the build tool IBM ASM Compiler do the following:
 - a In the navigation area select ASMPARM.
 - b In the Build Options section of the content area click New Object.

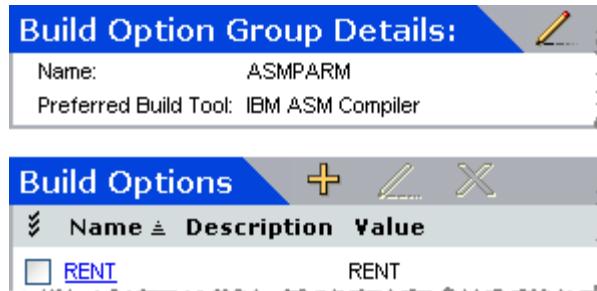


The Add New Build Option to Group dialog box appears.



- c For **Name** type *RENT*.
- d From the **Value Type** list select Option Rule.
- e From the **Build Tool** list select IBM ASM Compiler.
- f From the **Option Rule** list select Re-Entrant.

- g** From the **Value** list select RENT.
- h** Click OK. RENT is added to the list of build options for ASMPARM. Your content area should look similar to this:



To create the build option group LPARM:

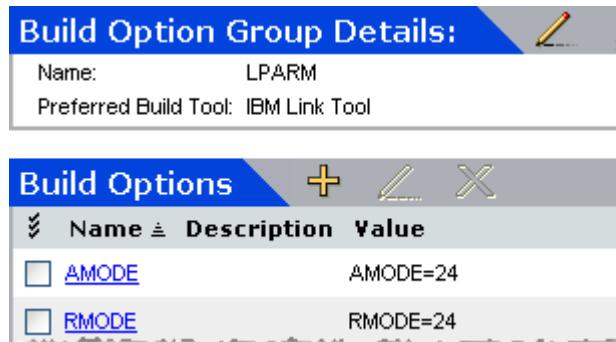
- 1** In the navigation area select Build Option Groups.
- 2** In the content or menu areas click New Object.



The Add New Build Option Group dialog box appears.

- 3** From the **Option Group Build Tool** list select IBM Link Tool.
- 4** For **Option Group Name** type *LPARM*.
- 5** Click OK. The new option group LPARM is added to the list of build option groups in the navigation and content areas.
- 6** To add the AMODE option from the build tool IBM Link Tool do the following:
 - a** In the navigation area select LPARM.
 - b** In the Build Options section of the content area click New Object. The Add New Build Option Group Option dialog box appears.
 - c** For **Name** type *AMODE*.
 - d** From the **Value Type** list select Option Rule.
 - e** From the **Build Tool** list select IBM Link Tool.
 - f** From the **Option Rule** list select Addressing Mode.
 - g** From the **Value** list select AMODE=24.
 - h** Click OK. AMODE is added to the list of build options for LPARM.
- 7** To add the RMODE option from the build tool IBM Link Tool do the following:
 - a** In the Build Options section of the content area click New Object. The Add New Build Option Group Option dialog box appears.
 - b** For **Name** type *RMODE*.
 - c** From the **Value Type** list select Option Rule.

- d** From the **Build Tool** list select IBM Link Tool.
- e** From the **Option Rule** list select Residence Mode.
- f** From the **Value** list select RMODE=24.
- g** Click OK. RMODE is added to the list of build options for LPARM. Your content area should look similar to this:



Exercise 10 Create Transition Rule Templates

In this exercise you will use Dimensions Build to create transition rule templates. A transition rule template describes a generic rule for building an item and applying it to specific source items at a later time. It describes how to change or transition an item of one file type, for example *.c, to an item of another file type, for example *obj, using a template or script. The purpose of transition rule templates is to save time when you are configuring build target definitions and to avoid having to manually enter definitions for each build target. Once you have set up a transition rule template you can reuse it as many times as necessary. You can create as many transition rule templates as you require. For more details see [page 169](#).

In this exercise you will create the following transition rule templates:

- *ASM Compile*—compiles an intermediate OBJ target from an ASM file using the pre-defined template TEMPLATE (MDHBASC0).
- *General Link*—compiles a final LOAD build target from an intermediate OBJ target and a SYSLIN file using the pre-defined template TEMPLATE (MDHBLNK1).

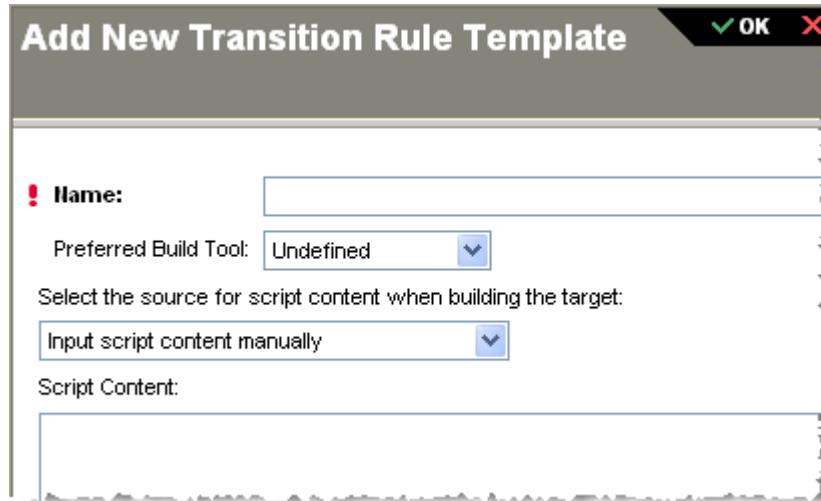
You will also assign the build option DMMAXRC to each transition rule template. DMMAXRC specifies the upper limit of non-zero values that will be accepted on each build step. You will specify a value of '4' so that the values 1, 2, 3, and 4 will be accepted.

To create the transition rule template **ASM Compile**:

- 1** In the navigation area of the Build Management tab click Settings and click Transition Rule Templates.
- 2** In the content or menu areas click New Object.



The Add New Transition Rule Template dialog box appears.



- 3 For **Name** type *ASM Compile*.
- 4 From the **Preferred Build Tool** list select *Undefined*.
- 5 From the **Select the source script content** list select *Use a file in the build area as a script*. Type the name of the following template: TEMPLATE(MDHBASC0)
- 6 Click OK. The transition rule template ASM Compile is added to the list of transition rule templates in the content area.
- 7 Check that ASM Compile is selected in the navigation area. In the content area select the Inputs tab and click New.



The Add New Transition Rule Template Input Mask dialog box appears. For **Mask Value** type *ASM(*)*. Click OK.

- 8 In the content area select the Outputs tab and click New.

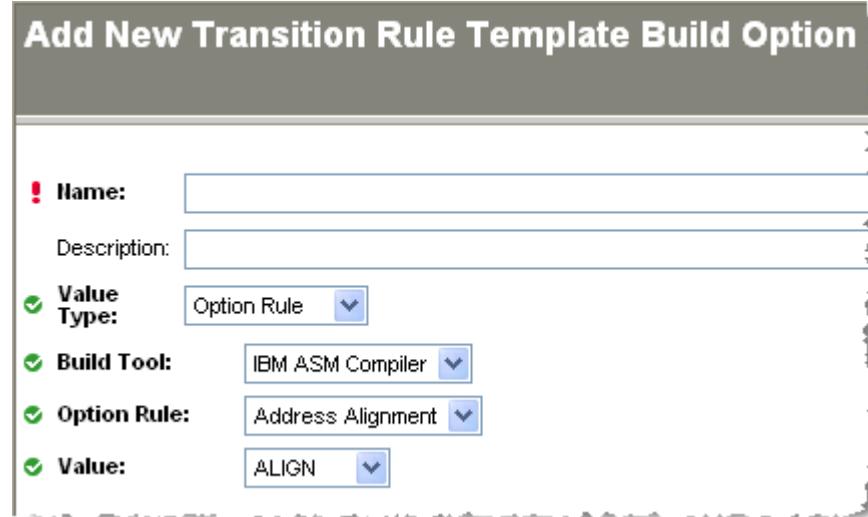


The Add New Transition Rule Template Output Mask dialog box appears. For **Mask Value** type *OBJ(*)*. Click OK.

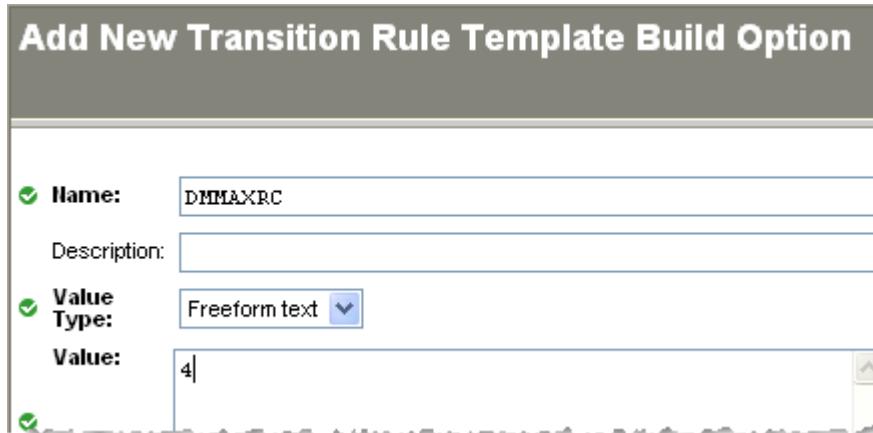
- 9 To add the build option DMMAXRC do the following:
 - a In the content area select the Build Options tab and click New.



The Add New Transition Rule Template Build Option dialog box appears.



- b For **Name** type DMMAXRC.
- c From the **Value Type** list select Freeform Text.
- d For **Value** type 4. Your dialog box should look like this:



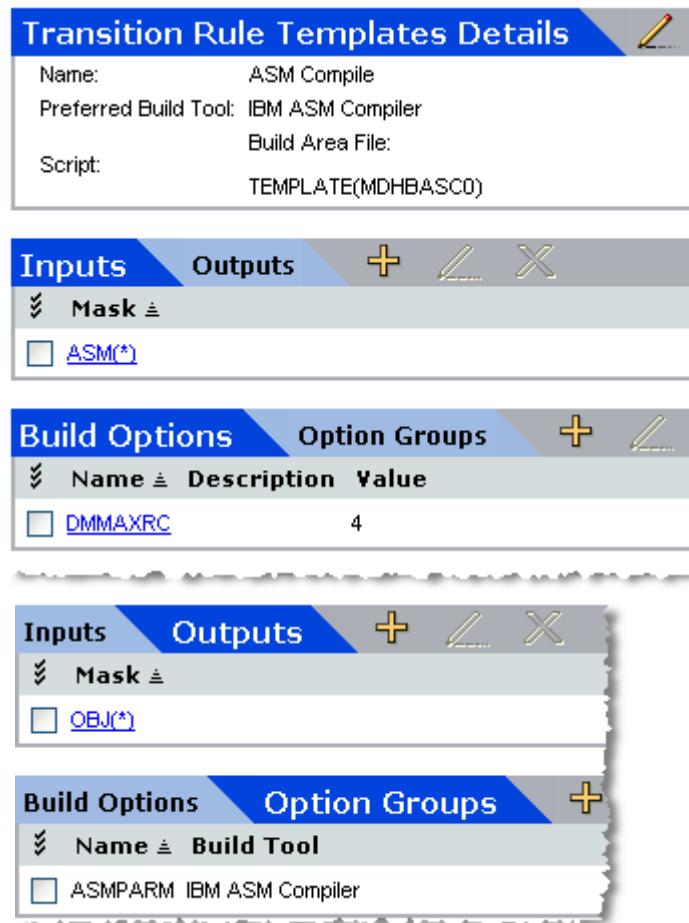
- e Click OK.
- 10 To add the build option group ASMPARM do the following:
 - a In the content area select the Option Groups tab and click New.



The Select Transition Rule Template Build Option Group dialog box appears.

- b From the list select ASMPARM.
- c Click OK.

The tabs for your ASM Compile transition rule template should look like this:



To create the transition rule template General Link:

- 1 In the navigation area of the Build Management tab click Settings and click Transition Rule Templates.
- 2 In the content or menu areas click New Object. The Add New Transition Rule Template dialog box appears.
- 3 For **Name** type *General Link*.
- 4 From the **Preferred Build Tool** list select *Undefined*.
- 5 From the **Select the source script content** list select *Use a file in the build area as a script*. Type the following: TEMPLATE(MDHBLNK1)
- 6 Click OK. The new transition rule template General Link is added to the list of transition rule templates in the content area.
- 7 Check that General Link is selected in the navigation area. In the content area select the Inputs tab and click New. The Add New Transition Rule Template Input Mask dialog box appears. For **Mask Value** type OBJ(*). Click OK.
- 8 On the Inputs tab click New again. For **Mask Value** type SYSLIN(*) and click OK.

- 9 In the content area select the Outputs tab and click New. The Add New Transition Rule Template Output Mask dialog box appears. For **Mask Value** type LOAD(*). Click OK.
- 10 To add the build option DMMAXRC do the following:
 - a In the content area select the Build Options tab and click New. The Add New Transition Rule Template Build Option dialog box appears.
 - b For **Name** type DMMAXRC.
 - c For **Value Type** list select Freeform Text.
 - d For **Value** type 4.
 - e Click OK.
- 11 To add the build option group LPARM do the following:
 - a In the content area select the Option Groups tab and click New. The Select Transition Rule Template Build Option Group dialog box appears.
 - b From the list select LPARM.
 - c Click OK.

The tabs for your General Link transition rule template should look like this:

Transition Rule Templates Details

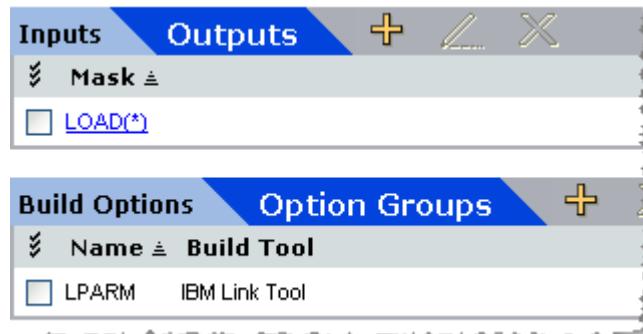
Name:	General Link
Preferred Build Tool:	IBM Link Tool
Build Area File:	
Script:	TEMPLATE(MDHBLNK1)

Inputs

Mask
<input type="checkbox"/> OBJ(*)
<input type="checkbox"/> SYSLIN(*)

Build Options

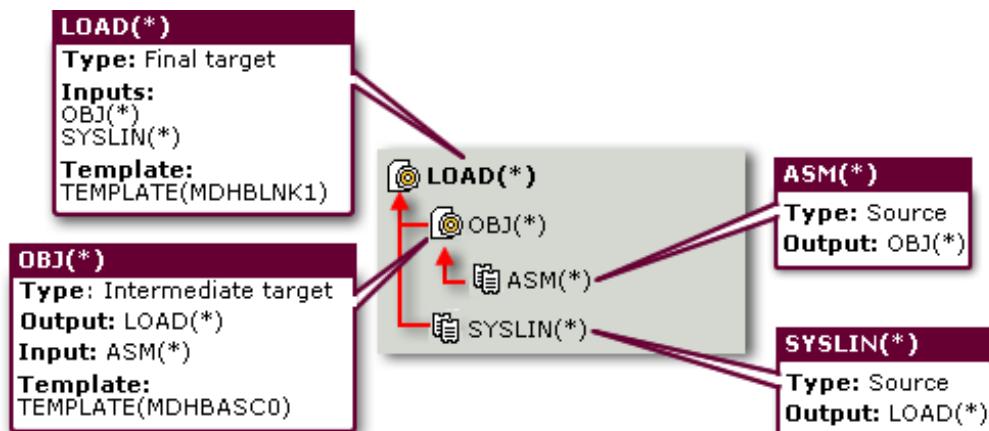
Name	Description	Value
<input type="checkbox"/> DMMAXRC		4



Exercise 11 Create an Application Rule Template

In this exercise you will use Dimensions Build to create an application rule template. An application rule template defines a logical group of transition rule templates that are used to create an application. For more details see [page 179](#).

In this exercise you will create an application rule template called ASM Program and add the transition rule templates that you created in the previous exercise. The generic build plan for the application rule template ASM Program looks like this:



To create the application rule template ASM Program:

- 1 In the navigation area of the Build Management tab click Settings and click Application Rule Templates.
- 2 In the content or menu areas click New Object.



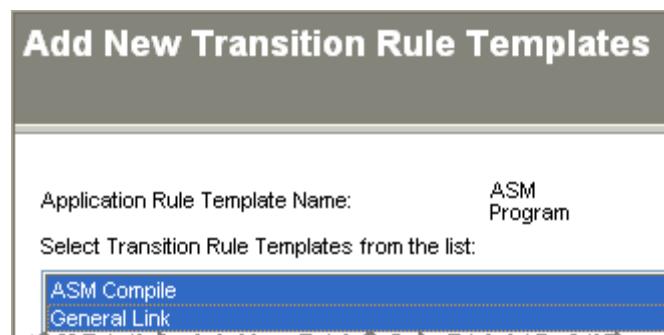
The Add New Transition Rule Template dialog box appears.

- 3** For **Name** type *ASM Program*.
- 4** Click OK.
- 5** Check that *ASM Program* is selected in the navigation area. In the content area, on the Transition Rule Templates tab, select New Object.

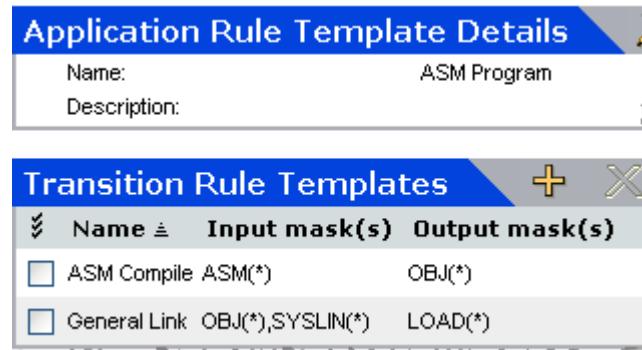


The Add New Transition Rule Templates dialog box appears.

- 6** From the **Select Transition Rule Templates** list select *ASM Compile* and *General Link*.



- 7** Click OK. The application rule template *ASM Program* should now look like this:



Note that the inputs and output masks are the same as those in the build plan above.

Exercise 12 Create the Build Targets

In this exercise you will use the Create New Target from Template wizard in Dimensions Build to create a build plan for each of the LOAD modules that you are going to build.

This exercise contains the following procedures:

- Creating the build target LOAD(MDOLRBLD), see [page 95](#).
- Creating the build target LOAD(MDOLRCHK), see [page 98](#).
- Creating the build target LOAD(MDOLRSRC), see [page 101](#).

The Create New Target from Template wizard is where you define your build targets. The wizard has three pages:

Target Page

Use the Target page to specify the properties of the target that you are going to build.

Name:	<input type="text"/>
File:	<input type="text"/>
Relative Path:	<input type="text"/>
Description:	<input type="text"/>
Is Virtual:	<input type="checkbox"/>
Is Final:	<input checked="" type="checkbox"/>
Design Part:	<input type="text"/>
Select Application Rule Template Please select template here.	

Sources Page

Use the Sources page to specify the sources that will be used to build the target.

Select the action from the following list:

Done
 Add new source
 Add existing source
 Modify selected source
 Remove selected source(s)

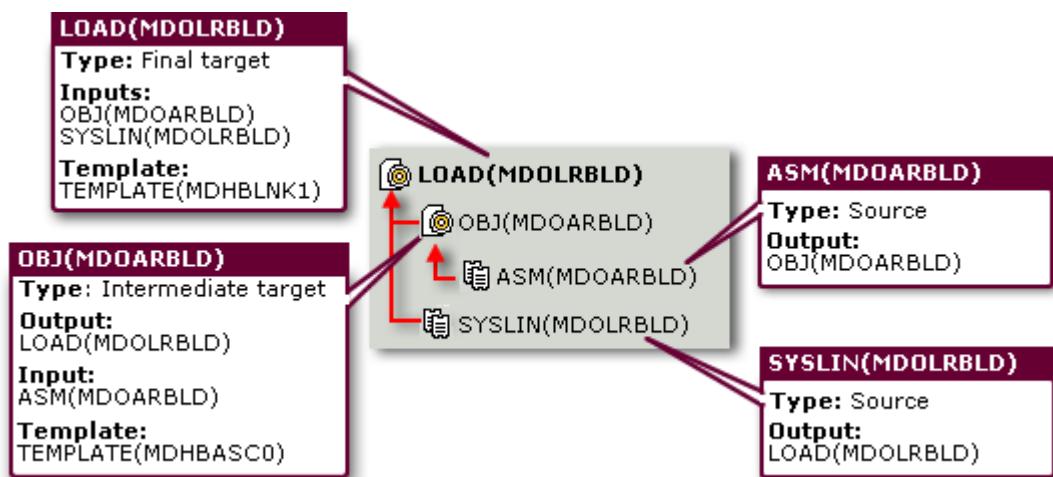
Confirmation Page

Use the Confirmation page to view and confirm that rules and transitions that will be used to build the target.

Target Sources Confirmation	The following application rule transitions will be generated: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">Input(s)</th><th style="text-align: left; padding: 2px;">Transition Name</th><th style="text-align: left; padding: 2px;">Output(s)</th></tr> </thead> <tbody> <tr> <td style="padding: 2px;">ASM(MDOARBLD)</td><td style="padding: 2px;">ASM Compile</td><td style="padding: 2px;">OBJ(MDOARBLD)</td></tr> <tr> <td style="padding: 2px;">OBJ(MDOARBLD)</td><td style="padding: 2px;">General Link</td><td style="padding: 2px;">LOAD(MDOARBLD)</td></tr> </tbody> </table>			Input(s)	Transition Name	Output(s)	ASM(MDOARBLD)	ASM Compile	OBJ(MDOARBLD)	OBJ(MDOARBLD)	General Link	LOAD(MDOARBLD)
Input(s)	Transition Name	Output(s)										
ASM(MDOARBLD)	ASM Compile	OBJ(MDOARBLD)										
OBJ(MDOARBLD)	General Link	LOAD(MDOARBLD)										

To create the build target for LOAD(MDOLRBLD):

When you have completed this procedure your build target will look like this:



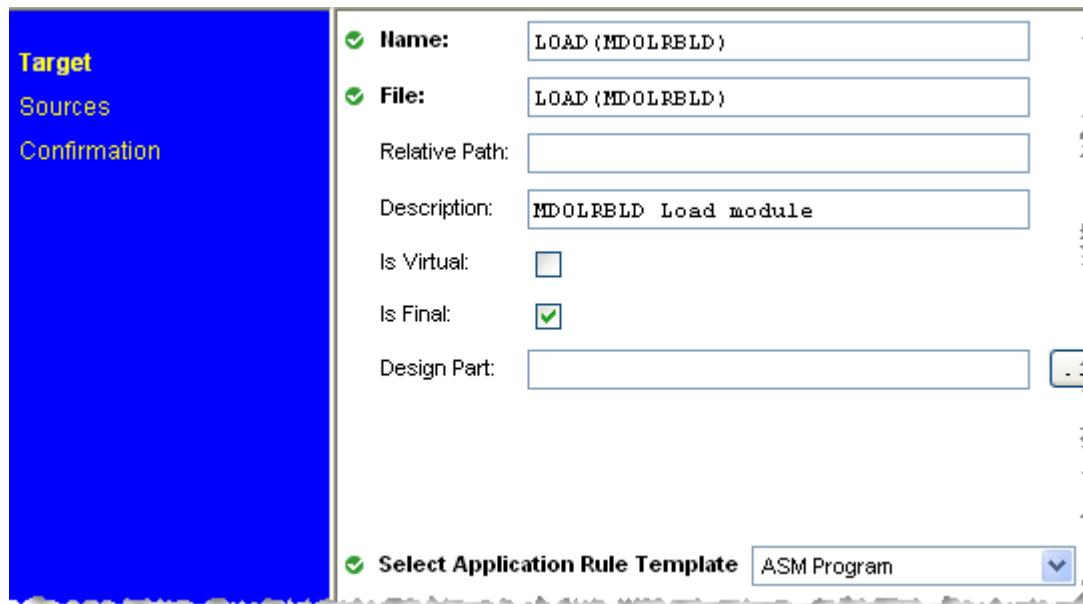
- 1 In the navigation area of the Build Management tab select the build project Disassembler, select the build configuration Disassembler, and select Build Targets.
- 2 In the content area click the Build Targets tab and click Define New Targets Using Templates.



The Create New Target wizard appears.

- 3 On the Targets page, to specify the final build target LOAD(MDOLRBLD), do the following:
 - a For **Name** type *LOAD(MDOLRBLD)*.
 - b Press your tab key. The target name is copied to the **File** field.
 - c Leave the **Relative Path** field empty.
 - d For **Description** type *MDOLRBLD Load module*.

- e To specify that LOAD(MDOLRBLD) is the final target for this build plan, select the **Is Final** check box.
- f From the **Select Application Rule Template** list select *ASM Program*.



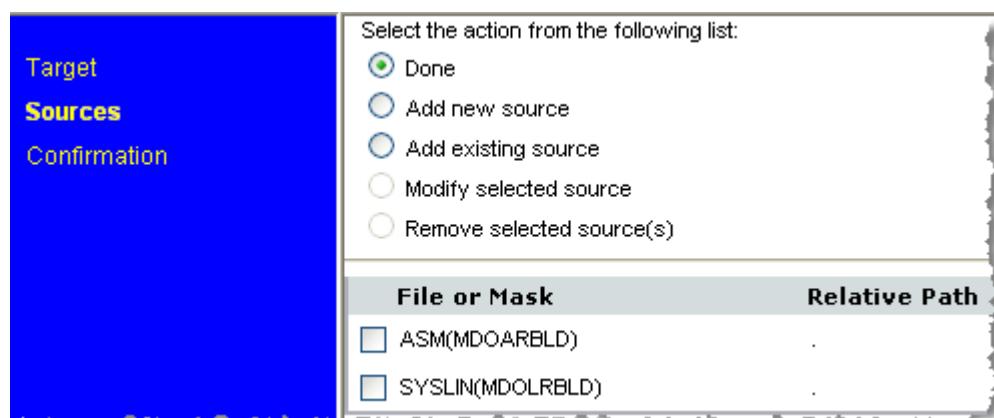
- 4 Click Next. The Scripts page appears.
- 5 To add the input ASM(MDOARBLD) do the following:
 - a Select Add New Source.

Select the action from the following list:

 - Done
 - Add new source
 - Add multiple

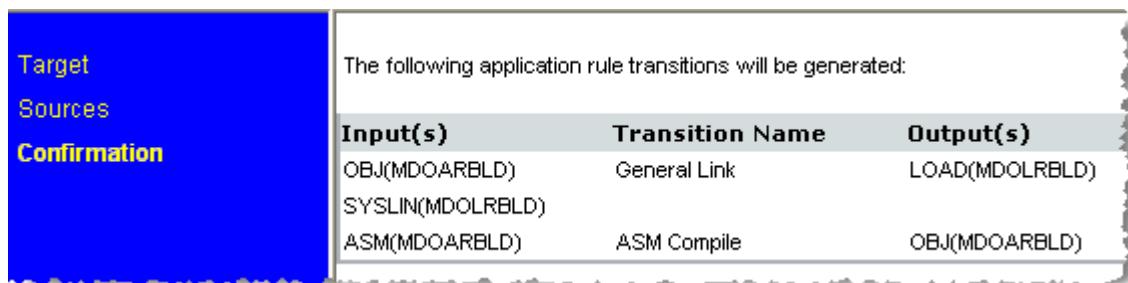
 Click Next.
 - b In the Add New Source page, for **File Name or Mask** type ASM(MDOARBLD).
 - c For **Relative Path** type a dot (.)
 - d Click Next.
- 6 To add the input SYSLIN(MDOARBLD) do the following:
 - a On the Scripts page select Add New Source and click Next.
 - b In the Add New Source page, for **File Name or Mask** type SYSLIN(MDOLRBLD).

- c For **Relative Path** type a dot (.).
- d Click Next. Your Sources page should look like this:



You do not need to define the intermediate target OBJ(MDOARBLD) as it is created from the source ASM(MDOARBLD).

- 7 At the bottom of the Sources page click Next.
- 8 The Confirmation page displays a list of all the outputs that will be generated, the inputs, and the transition rule templates used to create the outputs. Your Confirmation page should look like this:

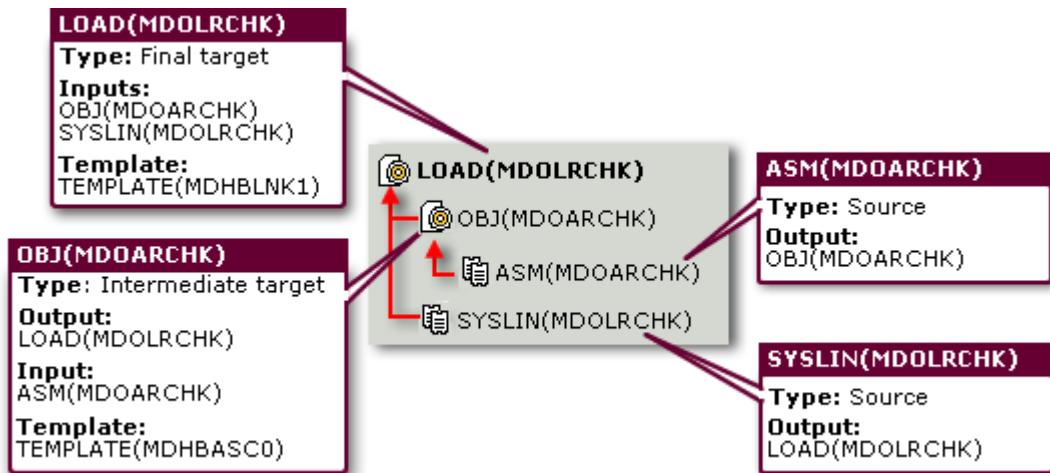


- 9 Click Finish. The build target LOAD(MDOLRBLD) is added to the build configuration. Note that the hierarchy of LOAD(MDOLRBLD) in the navigation area is identical to the build target diagram at the start of this procedure.



To create the build target for LOAD(MDOLRCHK):

When you have completed this procedure your build target will look like this:



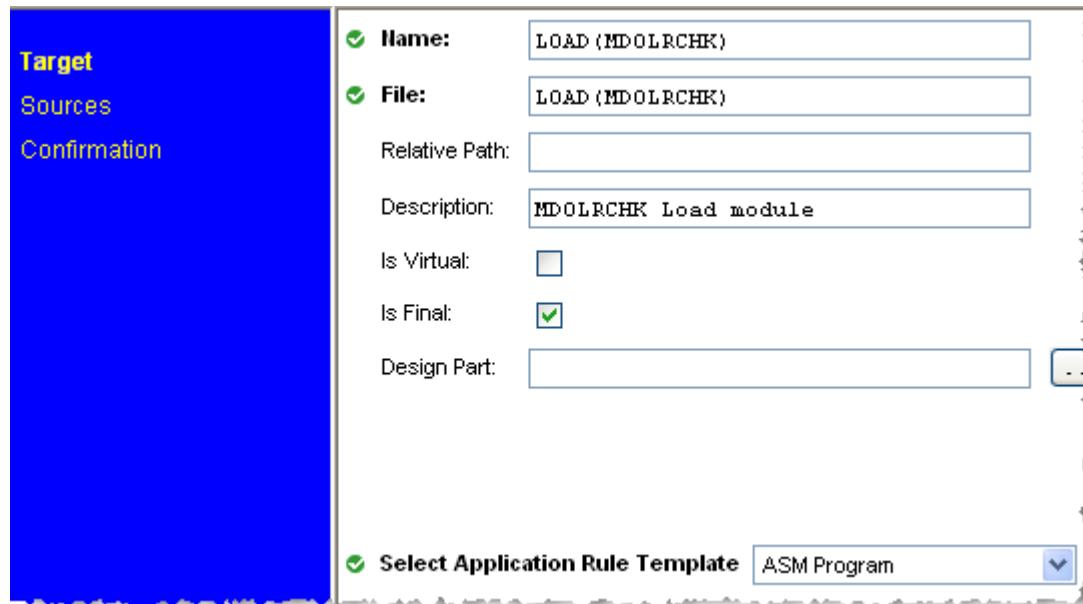
- 1 In the navigation area of the build configuration Disassembler select Build Targets.
- 2 In the content area click the Build Targets tab and click Define New Targets Using Templates.



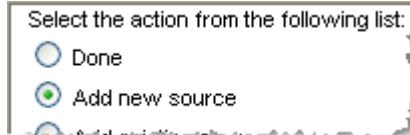
The Create New Target wizard appears.

- 3 On the Targets page, to specify the final build target **LOAD(MDOLRCHK)**, do the following:
 - a For **Name** type *LOAD(MDOLRCHK)*.
 - b Press your tab key. The target name is copied to the **File** field.
 - c Leave the **Relative Path** field empty.
 - d For **Description** type *MDOLRCHK Load module*.
 - e To specify that **LOAD(MDOLRCHK)** is the final target for this build plan, select the **Is Final** check box.

- f From the **Select Application Rule Template** list select *ASM Program*.



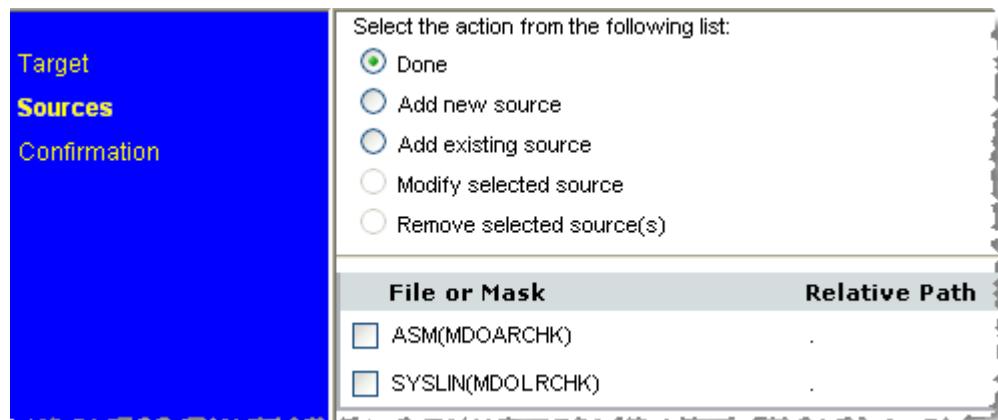
- 4 Click Next. The Scripts page appears.
- 5 To add the input ASM(MDOARCHK) do the following:
- On the Scripts page select Add New Source.



Click Next.

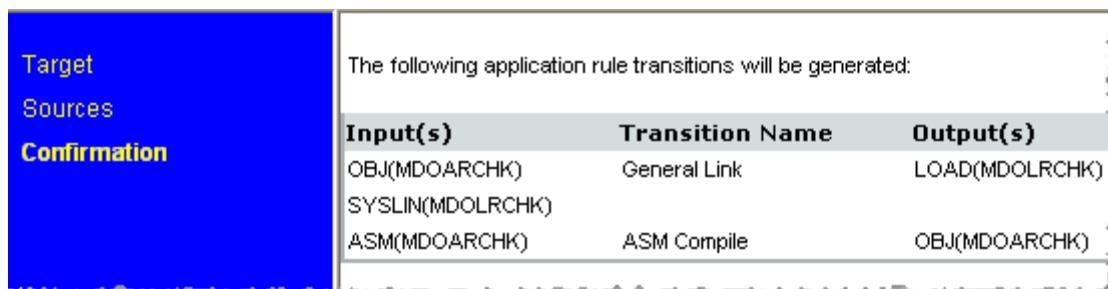
- In the Add New Source page, for **File Name or Mask** type ASM(MDOARCHK).
 - For **Relative Path** type a dot (.).
 - Click Next.
- 6 To add the input SYSLIN(MDOLRCHK) do the following:
- On the Scripts page select Add New Source and click Next.
 - In the Add New Source page, for **File Name or Mask** type SYSLIN(MDOLRCHK).
 - For **Relative Path** type a dot (.).

- d Click Next. Your Sources page should look like this:



You do not need to define the intermediate target OBJ(MDOARCHK) as it is created from the source ASM(MDOARCHK).

- 7 At the bottom of the Sources page click Next.
- 8 The Confirmation page displays a list of all the outputs that will be generated, the inputs, and the transition rule templates used to create the outputs. Your Confirmation page should look like this:

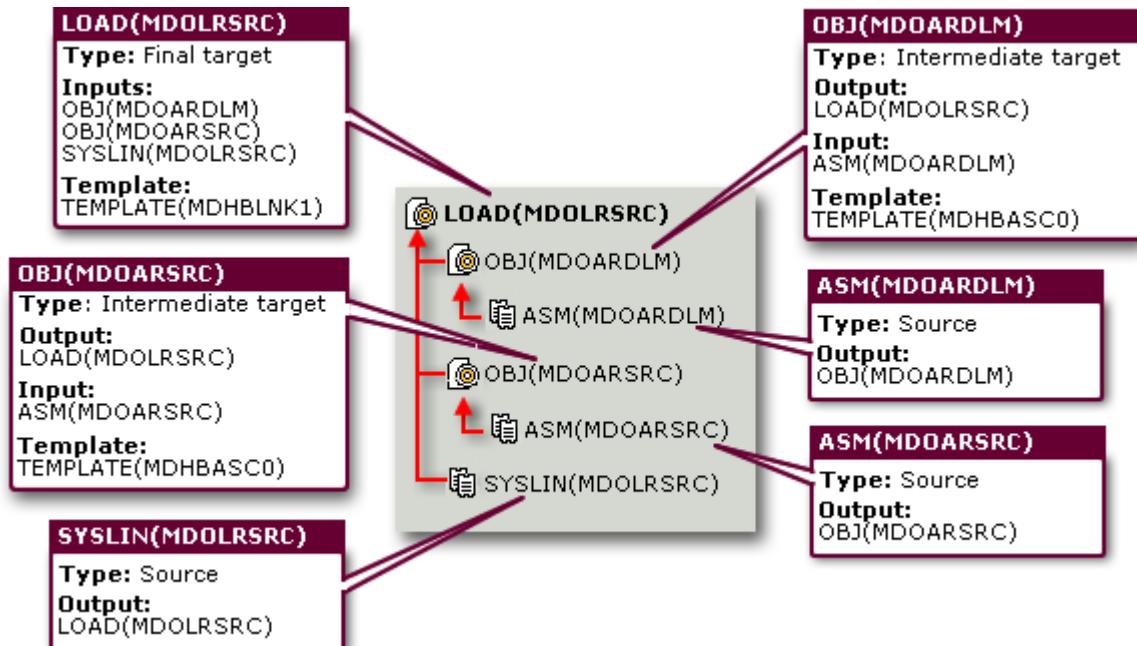


- 9 Click Finish. The build target LOAD(MDOLRCHK) is added to the build configuration. Note that the hierarchy of LOAD(MDOLRCHK) in the navigation area is identical to the build target diagram at the start of this procedure.



To create the build target for LOAD(MDOLRSRC):

When you have completed this procedure your build target will look like this:



NOTE The target **LOAD(MDOLRSRC)** has two intermediate targets, **OBJ(MDOARSRC)** and **OBJ(MDOARDLM)**.

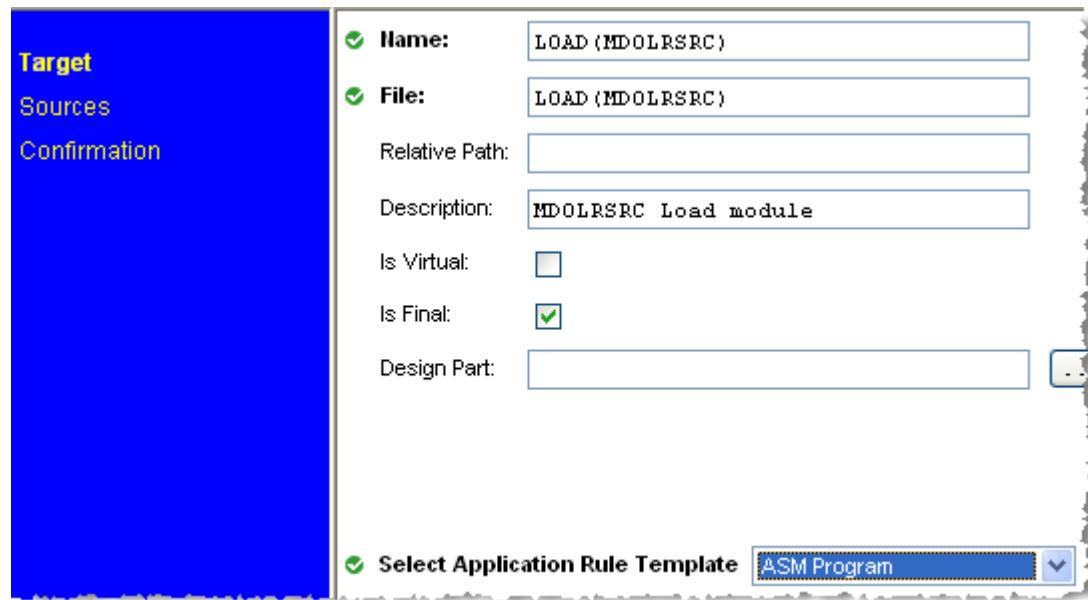
- 1 In the navigation area of the Build Management tab expand the build project Disassembler, expand the build configuration Disassembler, and select Build Targets.
- 2 In the content area click the Build Targets tab and click Define New Targets Using Templates.



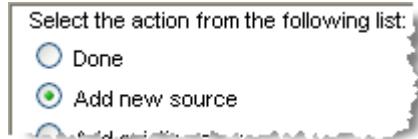
The Create New Target wizard appears.

- 3 On the Targets page, to specify the final build target **LOAD(MDOLRSRC)**, do the following:
 - a For **Name** type **LOAD(MDOLRSRC)**.
 - b Press your tab key. The target name is copied to the **File** field.
 - c Leave the **Relative Path** field empty.
 - d For **Description** type **MDOLRSRC Load module**.
 - e To specify that **LOAD(MDOLRSRC)** is the final target for this build plan, select the **Is Final** check box.

- f From the **Select Application Rule Template** list select *ASM Program*.



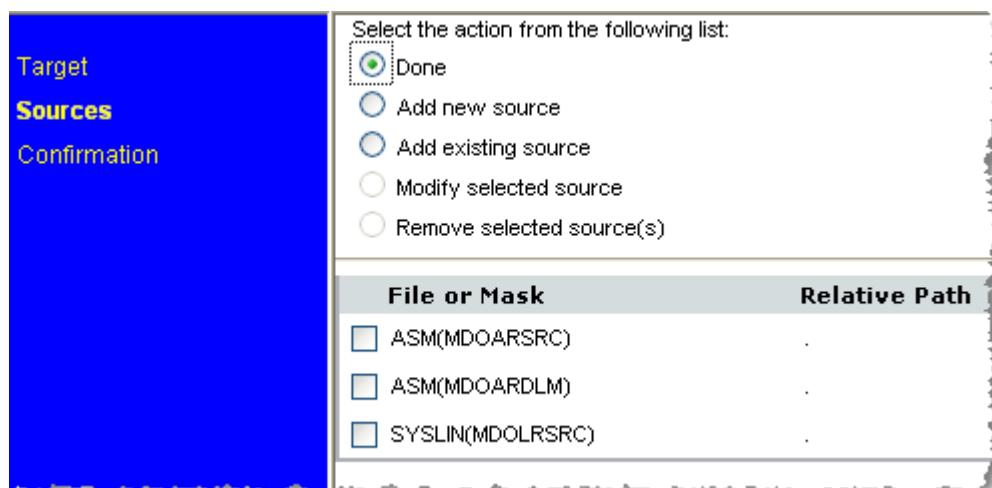
- 4 Click Next. The Scripts page appears.
- 5 To add the input ASM(MDOARSRC) do the following:
- Select Add New Source.



Click Next.

- In the Add New Source page, for **File Name or Mask** type ASM(MDOARSRC).
 - For **Relative Path** type a dot (.).
 - Click Next.
- 6 To add the input ASM(MDOARDLM) do the following:
- On the Scripts page select Add New Source.
- Click Next.
- In the Add New Source page, for **File Name or Mask** type ASM(MDOARDLM).
 - For **Relative Path** type a dot (.).
 - Click Next.
- 7 To add the input SYSLIN(MDOLRSRC) do the following:
- On the Scripts page select Add New Source and click Next.
 - In the Add New Source page, for **File Name or Mask** type SYSLIN(MDOLRSRC).

- c For **Relative Path** type a dot (.).
- d Click Next. Your Sources page should look like this:

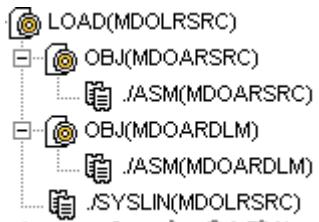


You do not need to define the intermediate targets OBJ(MDOARSRC) and OBJ(MDOARDLM) as they are created from the sources ASM(MDOARSRC) and ASM(MDOARDLM).

- 8 At the bottom of the Sources page click Next.
- 9 The Confirmation page displays a list of all the outputs that will be generated, the inputs, and the transition rule templates used to create the outputs. Your Confirmation page should look like this:

Target	The following application rule transitions will be generated:		
Sources	Input(s)	Transition Name	Output(s)
Confirmation	OBJ(MDOARSRC)	General Link	LOAD(MDOLRSRC)
	OBJ(MDOARDLM)		
	SYSLIN(MDOLRSRC)		
	ASM(MDOARSRC)	ASM Compile	OBJ(MDOARSRC)
	ASM(MDOARDLM)	ASM Compile	OBJ(MDOARDLM)

- 10 Click Finish. The build target LOAD(MDOLRSRC) is added to the build configuration. Note that the hierarchy of LOAD(MDOLRSRC) in the navigation area is identical to the build target diagram at the start of this procedure.



Exercise 13 Check in the Build Configuration

In this exercise you will use Dimensions Build to check in the Disassembler build configuration. You must check in the build configuration before you can build it.

To check in the build configuration:

- 1 In the navigation area select the build configuration Disassembler. The icon with the red check mark  indicates that the configuration is checked out.
- 2 In the menu area click Checkin. The Provide Check-In Comment dialog box appears.
- 3 For **Comment** type *rev1.0*.
- 4 Click OK. The icon for Disassembler changes to one without a red check mark  indicating that the build configuration is checked in.
- 5 In the navigation area click Versions. The content area refreshes and displays information similar to the following:

Versions				
Version	Created	Creator	Comment	
1	Tue Feb 07 2006 9:05 AM PST	dmsys	rev1.0	

Exercise 14 Run the Build

In this exercise you will use the Dimensions CM desktop client to build the Disassembler project.

To run the build:

- 1 Log in to the Dimensions CM desktop client, ask your administrator for the log in parameters.
- 2 Open the Disassembler project.
- 3 From the Project menu select Build Project. The Run Build for Project dialog box opens.
- 4 From the Configuration list select *Disassembler;0*.



NOTE The Configuration list displays the latest checked in version of the build configuration for the Disassembler project in the following format:

<build configuration name>;<version number>

Your build configuration version number may be different from the one in the illustration above.

- 5 Select the **Work area** option. From the Areas list select **DISASSEMBLER-GEN**

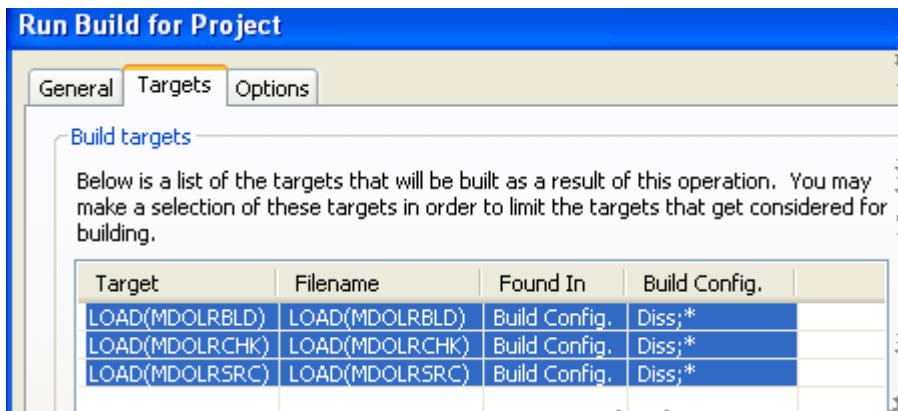


- 6 To capture the build outputs and check them automatically into Dimensions, select the **Capture build outputs** check box.

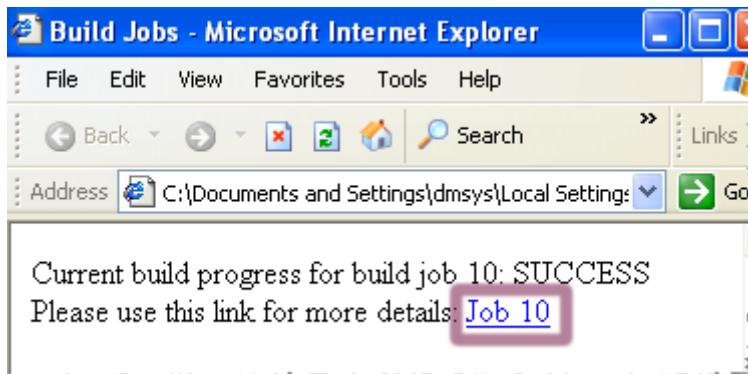


- 7 If your process model specifies that checked in items have to be related to a request, in the **Request(s)** field type a request ID, or click Browse and use the Find Request(s) dialog box to select a request.

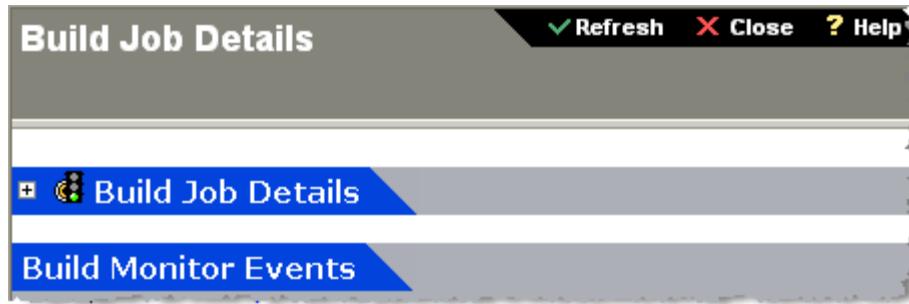
- 8 Click the Targets tab. The tab displays a list of the final targets that will be built and should look similar to this:



- 9 Click OK. After a short delay a browser window opens and displays a short description of the status of the build job. To view details of the build job click the link on the second line.



The browser refreshes and displays the Build Job Details window.



The **Build Job Details** section of the Build Job Details window has the following information (click the '+' button to expand the section):

A screenshot of the "Build Job Details" section expanded. It lists various parameters and their values:

Build Job ID:	46
Started at:	Thu Dec-07 1:52PM
Finished at:	Thu Dec-07 1:50PM
Initiated by:	dmsys
Build Result:	Build job executed successfully
Dimensions Project:	PAYROLL:DISASSEMBLER
Build Configuration:	Disassembler
Platform:	MVS
Targets:	LOAD(MDOLRBLD); OBJ(MDOARBLD); LOAD(MDOLRCHK); OBJ(MDOARCHI); LOAD(MDOLRSRC); OBJ(MDOARSRC); OBJ(MDOARDLM);

- The ID of the build job.
- The start and finish times of the build job.
- The Dimensions user who initiated the build.
- The result of the build.
- The Dimensions project containing the build configuration.
- The build configuration that you are building.
- The operating system platform.
- The targets that are being built.

The **Build Monitor Events** section of the Build Job Details window has the following columns:

Build Monitor Events						
Page: 12		Results				
Order	Step	Type	Time	Target	Task	Message
1		Build started	Thu Dec-07 1:48PM			
2	1	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD job(J0619969)	
3	2	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK job(J0619970)	
4	1	Task Build finished	Thu Dec-07 1:48PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD	rc=0
5	1	Information	Thu Dec-07 1:53PM	OBJ(MDOARBLD)		Process target
6	3	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARSRC)	S3 Assemble MDOARSRC job(J0619971)	
7	2	Task Build finished	Thu Dec-07 1:49PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK	rc=0

- **Order:** The order in which the tasks were started. Build tasks are run asynchronously therefore this is not the order of steps defined in the build template.
- **Step:** A synonym for a BRD (Build Request Definition) task or a single JCL step in a build template.
- **Type:** The type of task, for example, *Build started* or *Task Build finished*.
- **Time:** The time the task was started.
- **Target:** The name of the target being built.
- **Task:** A task in a BRD.
- **Message:** Any messages relevant to the task. The following types of messages are common:
 - job(<job number>): The mainframe job number.
 - Process targets deployment: the deployment status of the item.
 - rc=<n> where <n> is the number of the return code.
 - MDHnnnnnnnnE where E specifies that this is an error message.

To change the order in which tasks are displayed, click a heading at the top of a column. For example, to view the steps in sequential order, click the Task column heading.

Do not close the Build Job Details window, you will be using it in the next exercise.

Exercise 15 Monitor the Build

In this exercise you will use the Build Monitor Events window to monitor:

- The general progress of the build.
- The build of the final target LOAD(MDOLRBLD) and its intermediate target OBJ(MDOARBLD).

To monitor the build:

- 1 If your build was successful the Build Monitor Events section of the Build Job Details window should look similar to this:

Build Monitor Events						
Order	Step	Type	Time	Target	Task	Message
1		Build started	Mon Jan-29 1:16PM			
2	1	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD job(J069431)	
3	2	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK job(J069431)	
4	1	Task Build finished	Mon Jan-29 1:16PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD rc=0	
5	1	Information	Mon Jan-29 1:20PM	OBJ(MDOARBLD)		Process targ
6	3	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARDLM)	S3 Assemble MDOARDLM job(J069431)	
7	2	Task Build finished	Mon Jan-29 1:16PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK rc=0	
8	2	Information	Mon Jan-29 1:20PM	OBJ(MDOARCHK)		Process targ
9	4	Task Build started	Mon Jan-29 1:17PM	OBJ(MDOARSRC)	S4 Assemble MDOARSRC job(J069431)	
10	3	Task Build finished	Mon Jan-29 1:17PM	OBJ(MDOARDLM)	S3 Assemble MDOARDLM rc=0	
11	3	Information	Mon Jan-29 1:20PM	OBJ(MDOARDLM)		Process targ
12	5	Task Build started	Mon Jan-29 1:17PM	LOAD(MDOLRBLD)	S5 Link MDOLRBLD	job(J069432)
13	4	Task Build finished	Mon Jan-29 1:17PM	OBJ(MDOARSRC)	S4 Assemble MDOARSRC rc=4	
14	4	Information	Mon Jan-29 1:20PM	OBJ(MDOARSRC)		Process targ
15	6	Task Build started	Mon Jan-29 1:17PM	LOAD(MDOLRCHK)	S6 Link MDOLRCHK	job(J069432)
16	5	Task Build finished	Mon Jan-29 1:17PM	LOAD(MDOLRBLD)	S5 Link MDOLRBLD	rc=0
17	5	Information	Mon Jan-29 1:21PM	LOAD(MDOLRBLD)		Process targ
18	7	Task Build started	Mon Jan-29 1:17PM	LOAD(MDOLRSRC)	S7 Link MDOLRSRC	job(J069432)
19	6	Task Build finished	Mon Jan-29 1:17PM	LOAD(MDOLRCHK)	S6 Link MDOLRCHK	rc=0
20	6	Information	Mon Jan-29 1:21PM	LOAD(MDOLRCHK)		Process targ
21	7	Task Build finished	Mon Jan-29 1:17PM	LOAD(MDOLRSRC)	S7 Link MDOLRSRC	rc=0
22	7	Information	Mon Jan-29 1:21PM	LOAD(MDOLRSRC)		Process targ
23		Build finished	Mon Jan-29 1:17PM			



NOTE If there are errors or other problems in your build, the line numbers (in the Order column) associated with each task may be different to the ones in the illustration above.

- 2 Line 1 *Build started* is the time that you initiated the build job.

Build Monitor Events				
Order	Step	Type	Time	Target
1		Build started	Thu Dec-07 1:48PM	
2	1	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
3	2	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARCHK)
4	1	Task Build finished	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
5	1			

- 3 Line 2 *Task Build started* is the time that the build commenced for the intermediate target OBJ(MDOARBLD).

Build Monitor Events				
Order	Step	Type	Time	Target
1		Build started	Thu Dec-07 1:48PM	
2	1	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
3	2	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARCHK)
4	1	Task Build finished	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
5	1			

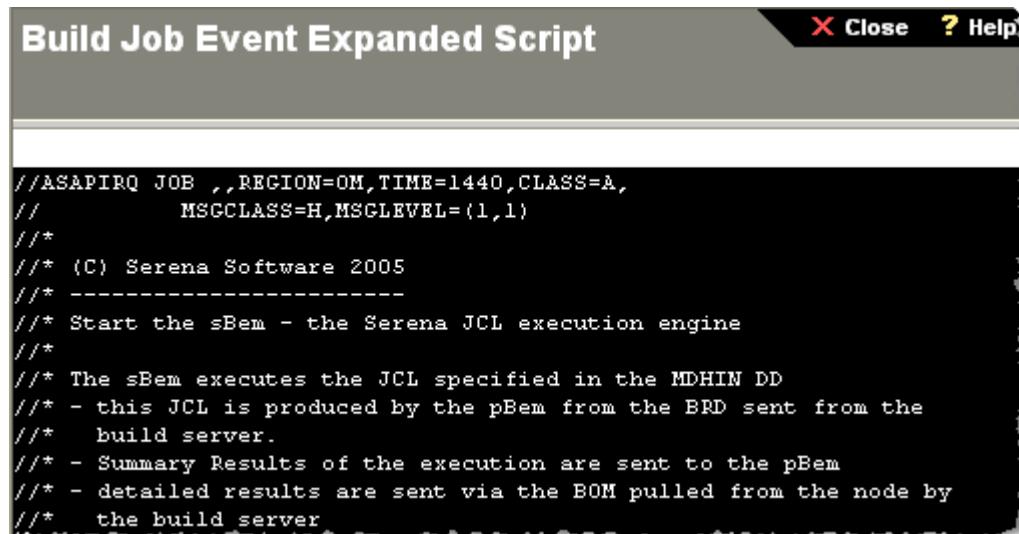
- 4 Line 4 *Task Build finished* is the time that the build finished for the intermediate target OBJ(MDOARBLD).

Build Monitor Events				
Order	Step	Type	Time	Target
1		Build started	Thu Dec-07 1:48PM	
2	1	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
3	2	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARCHK)
4	1	Task Build finished	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
5	1			

- 5 To view the expanded script for the build step for OBJ(MDOARBLD), in line four click the View Expanded Script icon.

Build Monitor Events				
Order	Step	Type	Time	Target
1		Build started	Thu Dec-07 1:48PM	
2	1	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
3	2	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARCHK)
4	1	Task Build finished	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
5	1			

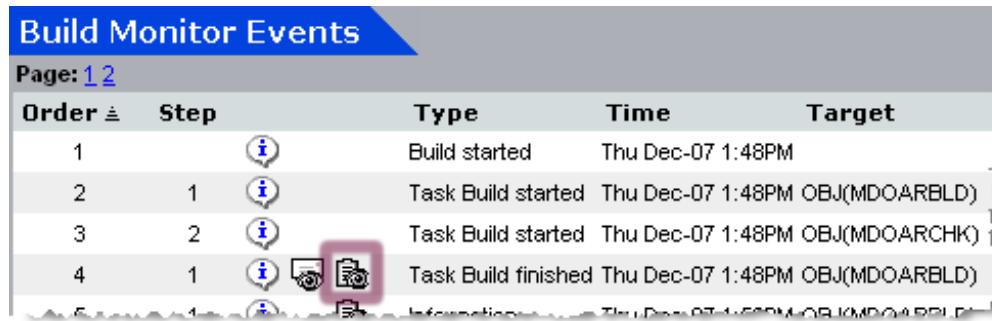
The Build Job Event Expanded Script dialog box appears and displays the content of the Serena mainframe Secondary Build Execution Monitor (SBEM) script.



```
//ASAPIRQ JOB ,REGION=OM,TIME=1440,CLASS=A,
//          MSGCLASS=H,MSGLEVEL=(1,1)
//*
//* (C) Serena Software 2005
*-----
/* Start the sBem - the Serena JCL execution engine
/*
/* The sBem executes the JCL specified in the MDHIN DD
/* - this JCL is produced by the pBem from the BRD sent from the
/* build server.
/* - Summary Results of the execution are sent to the pBem
/* - detailed results are sent via the BOM pulled from the node by
/* the build server
```

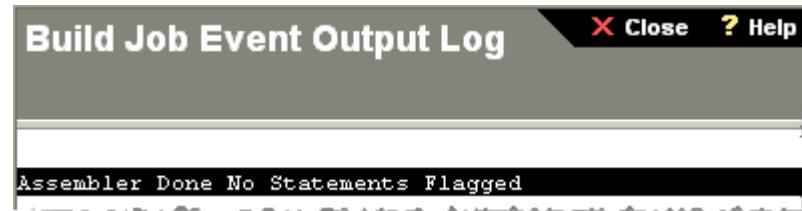
Close this dialog box when you have reviewed its content.

- To view build information for OBJ(MDOARBLD), in line 4 click View Output Log.



Build Monitor Events				
Order	Step	Type	Time	Target
1		Build started	Thu Dec-07 1:48PM	
2	1	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
3	2	Task Build started	Thu Dec-07 1:48PM	OBJ(MDOARCHK)
4	1	Task Build finished	Thu Dec-07 1:48PM	OBJ(MDOARBLD)
5	1	Information	Thu Dec-07 1:48PM	OBJ(MDOARBLD)

The Build Job Event Output Log dialog box appears. If the intermediate target OBJ(MDOARBLD) was built successfully the following message is displayed:

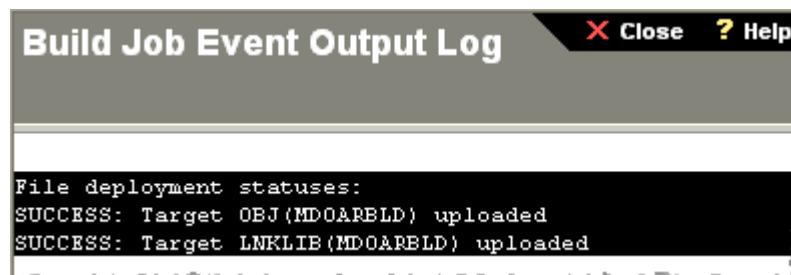


Close this dialog box when you have reviewed its content.

- 7 To view the target deployment information for OBJ(MDOARBLD), in line five *Information* click View Output Log.

Build Monitor Events				
Order	Step	Type	Time	Target
1		Build started	Fri Dec-08 9:05AM	
2	1	Task Build started	Fri Dec-08 9:05AM	OBJ(MDOARBLD)
3	2	Task Build started	Fri Dec-08 9:06AM	OBJ(MDOARCHK)
4	1	Task Build finished	Fri Dec-08 9:06AM	OBJ(MDOARBLD)
5	1	Information	Fri Dec-08 9:11AM	OBJ(MDOARBLD)

The Build Job Event Output Log dialog box appears. If the intermediate target OBJ(MDOARBLD) was uploaded successfully to Dimensions the following message is displayed:



Close this dialog box when you have reviewed its content.

- 8** Line 12 *Task Build started* is the time that the build commenced for the target LOAD(MDOLRBLD).

Build Monitor Events					
Order	Step	Type	Time	Target	Task
1		Build started	Mon Jan-29 1:16PM		
2	1	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD j
3	2	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK j
4	1	Task Build finished	Mon Jan-29 1:16PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD r
5	1	Information	Mon Jan-29 1:20PM	OBJ(MDOARBLD)	P
6	3	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARDLM)	S3 Assemble MDOARDLM j
7	2	Task Build finished	Mon Jan-29 1:16PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK r
8	2	Information	Mon Jan-29 1:20PM	OBJ(MDOARCHK)	P
9	4	Task Build started	Mon Jan-29 1:17PM	OBJ(MDOARSRC)	S4 Assemble MDOARSRC j
10	3	Task Build finished	Mon Jan-29 1:17PM	OBJ(MDOARDLM)	S3 Assemble MDOARDLM r
11	3	Information	Mon Jan-29 1:20PM	OBJ(MDOARDLM)	P
12	5	Task Build started	Mon Jan-29 1:17PM	LOAD(MDOLRBLD)	S5 Link MDOLRBLD j
13	4	Task Build finished	Mon Jan-29 1:17PM	OBJ(MDOARSRC)	S4 Assemble MDOARSRC r
14	4	Information	Mon Jan-29 1:20PM	OBJ(MDOARSRC)	P
15	6	Task Build started	Mon Jan-29 1:17PM	LOAD(MDOLRCHK)	S6 Link MDOLRCHK j
16	5	Task Build finished	Mon Jan-29 1:17PM	LOAD(MDOLRBLD)	S5 Link MDOLRBLD r
17	5	Information	Mon Jan-29 1:21PM	LOAD(MDOLRBLD)	P

- 9** Line 16 *Task Build finished* is the time that the build finished for the target LOAD(MDOLRBLD).

Build Monitor Events					
Order	Step	Type	Time	Target	Task
1		Build started	Mon Jan-29 1:16PM		
2	1	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD
3	2	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK
4	1	Task Build finished	Mon Jan-29 1:16PM	OBJ(MDOARBLD)	S1 Assemble MDOARBLD
5	1	Information	Mon Jan-29 1:20PM	OBJ(MDOARBLD)	P
6	3	Task Build started	Mon Jan-29 1:16PM	OBJ(MDOARDLM)	S3 Assemble MDOARDLM
7	2	Task Build finished	Mon Jan-29 1:16PM	OBJ(MDOARCHK)	S2 Assemble MDOARCHK
8	2	Information	Mon Jan-29 1:20PM	OBJ(MDOARCHK)	P
9	4	Task Build started	Mon Jan-29 1:17PM	OBJ(MDOARSRC)	S4 Assemble MDOARSRC
10	3	Task Build finished	Mon Jan-29 1:17PM	OBJ(MDOARDLM)	S3 Assemble MDOARDLM
11	3	Information	Mon Jan-29 1:20PM	OBJ(MDOARDLM)	P
12	5	Task Build started	Mon Jan-29 1:17PM	LOAD(MDOLRBLD)	S5 Link MDOLRBLD
13	4	Task Build finished	Mon Jan-29 1:17PM	OBJ(MDOARSRC)	S4 Assemble MDOARSRC
14	4	Information	Mon Jan-29 1:20PM	OBJ(MDOARSRC)	P
15	6	Task Build started	Mon Jan-29 1:17PM	LOAD(MDOLRCHK)	S6 Link MDOLRCHK
16	5	Task Build finished	Mon Jan-29 1:17PM	LOAD(MDOLRBLD)	S5 Link MDOLRBLD
17	5	Information	Mon Jan-29 1:21PM	LOAD(MDOLRBLD)	P

- 10** On line 16 optionally view the expanded script and event output log for LOAD(MDOLRBLD).
- 11** To view the target deployment information for LOAD(MDOARBLD), in line 17 *Information* click View Output Log. If the target was uploaded successfully to Dimensions the following message is displayed:



- 12** The final line *Build finished* displays the time that the build was completed.
If you receive an error, to view a summary click View Error Log in the appropriate line.



The Build Job Event Error Log dialog box appears. Close this dialog box when you have reviewed its content.

- 13** If there are no error messages, on the mainframe browse the build area data set <HLQ>. <ILQ>. DEV.LOAD and verify that the following Load modules were built:
- MDOLRBLD
 - MDOLRCHK
 - MDOLRSRC



TIP You can use the Build Monitoring tab to:

- Monitor the progress of active build jobs that are currently running.
- View the attempts that you have made to launch builds.
- View the execution history of builds that you have launched previously.

Exercise 16 Verify that the Targets were Created in Dimensions

In this exercise you will use the desktop client to verify that the final and intermediate targets were created in Dimensions.

To verify that the targets were created in Dimensions:

- 1** In the desktop client, in the Disassembler project, click the LOAD directory. The contents of the directory should look similar to this:

Filename	Revision	Status
MDOLRBLD.LOAD	1	BUILT
MDOLRCHK.LOAD	1	BUILT
MDOLRSRC.LOAD	1	BUILT

Note that the Status column says BUILT indicating that the final LOAD targets were built successfully.

- 2 Click the OBJ directory. The contents of the directory should look similar to this:

Folders and Items (PAYROLL:DISASSEMBLER)			
Folders and Items		4 Related Items	
	Filename	Revision	Status
	MDOARBLD.OBJ	1	BUILT
	MDOARCHK.OBJ	1	BUILT
	MDOARDLM.OBJ	1	BUILT
	MDOARSRC.OBJ	1	BUILT

Note that the Status column states BUILT indicating that the intermediate OBJ targets were built successfully.



NOTE The following directories were created as part of the build process:

- LNKLIB
- ASMLIST
- LNK1LIST
- LNK2LIST

These are listing type data sets containing outputs from the build steps.

Exercise 17 View Derived Items

In this exercise you will use the desktop client to view some of the derived items for the Disassembler project. A derived items list can show the following:

- The item(s) used to build a target.
- The item(s) built from a source.



NOTE Before you can view derived items in desktop client you must customize your Items view so that it includes the Derived Items object class.

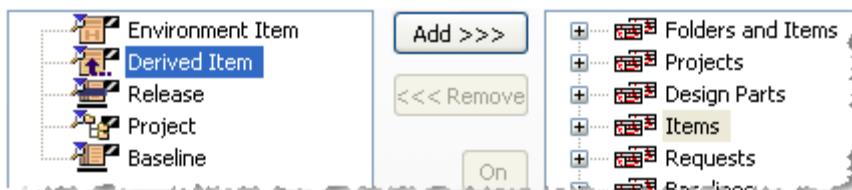
To customize your Items view see below.

To view the derived items for the target LOAD(MDOLRBLD) see [page 116](#).

To view the items derived from the source ASM(MDOARBLD) see [page 117](#).

To customize your Items view:

- 1 In desktop client, from the Tools menu select Customize Views. The Customize dialog box appears.
- 2 Click the Views tab.
- 3 In the right pane select Items.
- 4 In the left pane select Derived Items and click **Add >>>**.



- 5 In the right pane expand Items. Derived Items should now appear in the Items tree.



- 6 Click OK.
- 7 Restart the desktop client.

To view the derived items for the target LOAD(MDOLRBLD):

- 1 In the Disassembler project select the folder LOAD.
- 2 In the Items window right click the item MDOLRBLD.LOAD and select Open New Window.

- 3 In the new Items window, on the left select MDOLRBLD.LOAD.

The screenshot shows the 'Items' window with a single item listed:

	Filename	Revision	Status	Update Date	Specification
<input checked="" type="checkbox"/>	MDOLRBLD.LOAD	1	BUILT	07-Dec-2006 14:54:27	PAYROLL:MDOLRBLD

- 4 On the right select the Derived Items tab. The Derived Items tab has two items:
- MDOARBLD.OBJ: the intermediate target that was created from the source MDOARBLD.ASM.
 - MDOLRBLD.SYSLIN: a source file.

The screenshot shows the 'Derived Items' tab with two entries:

	Filename	Revision	Status	Relationship
<input type="checkbox"/>	MDOARBLD.OBJ	1	BUILT	Derived
<input type="checkbox"/>	MDOLRBLD.SYSLIN	1	UNDER WORK	Derived

The icon with the left pointing arrow next to each item indicates that the item was used to build the target MDOLRBLD.LOAD. Note also that the Relationship column has the status 'Derived'.

- 5 To view derived items for other targets in the folder LOAD repeat steps 2–4.

To view the items derived from the source ASM(MDOARBLD):

- In the Disassembler project select the folder ASM.
- In the Items window right click the item MDOLRBLD.ASM and select Open New Window.
- In the new Items window, on the left select MDOARBLD.ASM.

The screenshot shows the 'Items' window with one item listed:

	Filename	Revision	Status	Update Date	Specification
<input checked="" type="checkbox"/>	MDOARBLD.ASM	1	UNDER WORK	08-Dec-2006 09:29:01	PAYROLL:MDO

- 4** On the right select the Derived Items tab. The Derived Items tab has four items:
- MDOARBLD.ASMLIST: an output from the build steps.
 - MDOARBLD.LNK1LIST: an output from the build steps.
 - MDOARBLD.LINKLIB: an output from the build steps.
 - MDOARBLD.OBJ: an intermediate target.

	Filename	Revision	Status	Relationship
<input type="checkbox"/> MDOARBLD.ASMLIST	1	UNDER WORK	Derived	
<input type="checkbox"/> MDOARBLD.LNK1LIST	1	UNDER WORK	Derived	
<input type="checkbox"/> MDOARBLD.LINKLIB	1	BUILT	Derived	
<input type="checkbox"/> MDOARBLD.OBJ	1	BUILT	Derived	

The icon with the right pointing arrow next to each item indicates that the item was built from the source MDOARBLD.ASM. Note also that the Relationship column for each item has the status 'Derived'.

- 5** To view derived items for other sources in the folder ASM repeat steps 2–4.

Exercise 18 View the Bill of Materials

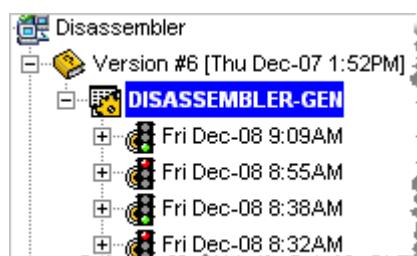
In this exercise you will use Dimensions Build to view the Bill of Materials (BOM) for the project.

To view the Bill of Materials:

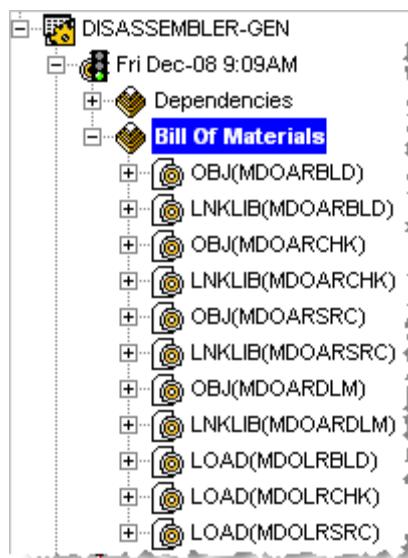
- 1 In Dimensions Build select the Build Monitoring tab.
- 2 In the navigation area click History, select PAYROLL-DISASSEMBLER, and select the Disassembler build configuration.



- 3 Select the last version of the Disassembler build configuration that you built successfully and select DISASSEMBLER-GEN.



- 4 Select the last successful build job (it should have a green traffic light) and select Bill of Materials.



The navigation area displays a complete BOM for the Disassembler project.

- 5 To view the BOM for a target, select it in the navigation area. The content area refreshes and displays the BOM for the item that you selected.

The screenshot shows the 'Bill of Materials: Target Details' interface. On the left, a navigation tree displays a hierarchy under 'DISASSEMBLER-GEN' and a date stamp 'Fri Dec-08 9:09AM'. Under 'Bill Of Materials', several items are listed, with 'OBJ(MDOARBLD)' highlighted in blue. The right side contains three main sections: 'Bill of Materials: Target Details' (with fields for Name, Path, Metadata, Build Type, Return Code, and Options), 'Dependent Targets' (listing none), and 'Dependent Sources' (listing various source components like ASM, MACLIB, LNKLIB, and LOAD).

- 6 To view the BOM for a source item, select it in the navigation area. The content area refreshes and displays information for the item that you selected.

The screenshot shows the 'BOM Source Details' interface. On the left, the navigation tree shows the same hierarchy as before, but 'ASM(MDOARBLD)' is now highlighted in blue. The right side contains two main sections: 'BOM Source Details' (with fields for Name, Path, and Metadata) and 'Dependent Sources' (listing none).

Summary

Congratulations, you have completed the Assembler MVS quick start tutorial. This tutorial has provided you with a quick overview of Dimensions Build and shown you how to do the following build-related tasks:

- Create a Dimensions project.
- Create mainframe items in the project.
- Add a build configuration to the Dimensions project.
- Create a build area and add it to the build configuration.
- Create build tools and build option groups.
- Create transition rule templates and an application rule template.
- Add build targets to the build configuration.
- Run and monitor the build job.
- View derived items.
- View the bill of materials.

To learn more about Dimensions Build concepts see [Chapter 4, "About Dimensions Build" on page 125](#).

Part 2

Overview of Dimensions Build

Part 2: Overview of Dimensions Build contains the following chapters

About Dimensions Build	125
Overview of the Build Administration User Interface	139

Chapter 4

About Dimensions Build

Introduction	126
What's New in Version 10.1.1?	127
Dimensions Build Terminology	127
The Build Flow	129
User Scenarios	131
Build Areas	133
Search Paths	135
Automatic Builds	136
Build Roles and Privileges	136
Capturing Build Outputs	136
Preservation Rules and Policies	137
Auditing Build Areas	138
Templates	138
Impacted Targets	138



NOTE For z/OS mainframes, the term directory refers to one or more MVS data set qualifiers at the start of data set names.

Introduction

Dimensions Build is a build management, execution, and monitoring tool that is part of Dimensions CM. Dimensions Build enables you to execute builds from the Build Administration cluster in the Dimensions CM administration console, or from the Dimensions desktop, web, and ISPF clients. For information about building from these clients see the *User's Guide* and the *Dimensions for z/OS User's and Administrator's Guide*. For information about the different ways that you can use Dimensions Build see "[User Scenarios](#)" on page 131.

Supported Platforms

Dimensions Build runs on all major Dimensions CM supported platforms: Windows, Solaris, HP-UX, IBM AIX, Linux, and z/OS mainframe (MVS and USS). For a complete list of supported platforms see the readme file that accompanies this release.

Build Engines

Dimensions Build is build engine independent and integrates with third-party engines on distributed and mainframe platforms.

On the distributed side build managers can use their preferred third-party build engine such as Ant or Serena ChangeMan Builder (Openmake). Dimensions Build integrates with those tools and can import Openmake target definition files and Ant XML build configuration files. For more information see [page 293](#).

On the mainframe side Dimensions Build natively supports the Serena mainframe build utility that is installed with Dimensions for z/OS. For more information see [page 323](#).

Versioning and Repeating Builds

You can create multiple versions of build configurations and repeat these builds whenever you want. Each version of a build configuration includes the following information:

- The target definitions including high-level dependencies.
- The templates used to build each target.
- The build area definitions (host, authentication details, file system location, etc.).

For more information see [page 183](#).

Scheduling Builds

You can schedule the execution of builds to suit your build paradigm. When you set up a scheduled build job you specify the build configuration and version that will be executed, the targets, the build area, and the start time. You can also specify the frequency at which a build reoccurs. For more information see [page 249](#).

Monitoring Builds

You can monitor the status of builds that are currently running and view the history of completed builds. For each build event you can view the expanded script used to build the step, the output log of link and compile listings for the target, and the error log (if applicable). For more information see [page 261](#).

Notifications

You can create and subscribe to e-mail notifications that update you about the progress of your build jobs. For more information see [page 277](#).

Integration with Dimensions CM

Dimensions CM performs the following functions for Dimensions Build:

- Drives the population of deployment areas.
- Authenticates and authorizes items.
- Preserves outputs and intermediate files generated by the build engines.
- Preserves bill-of-materials and post-build dependency information to enable you to perform impact analysis and traceability.
- Records which build configuration versions were in use at the time a baseline is taken.

Build Administration

Dimensions Build is administered in the Build Administration cluster of the Dimensions CM administration console. For more information about the user interface see [page 139](#).

What's New in Version 10.1.1?

The following features have been added to Dimensions Build version 10.1.1:

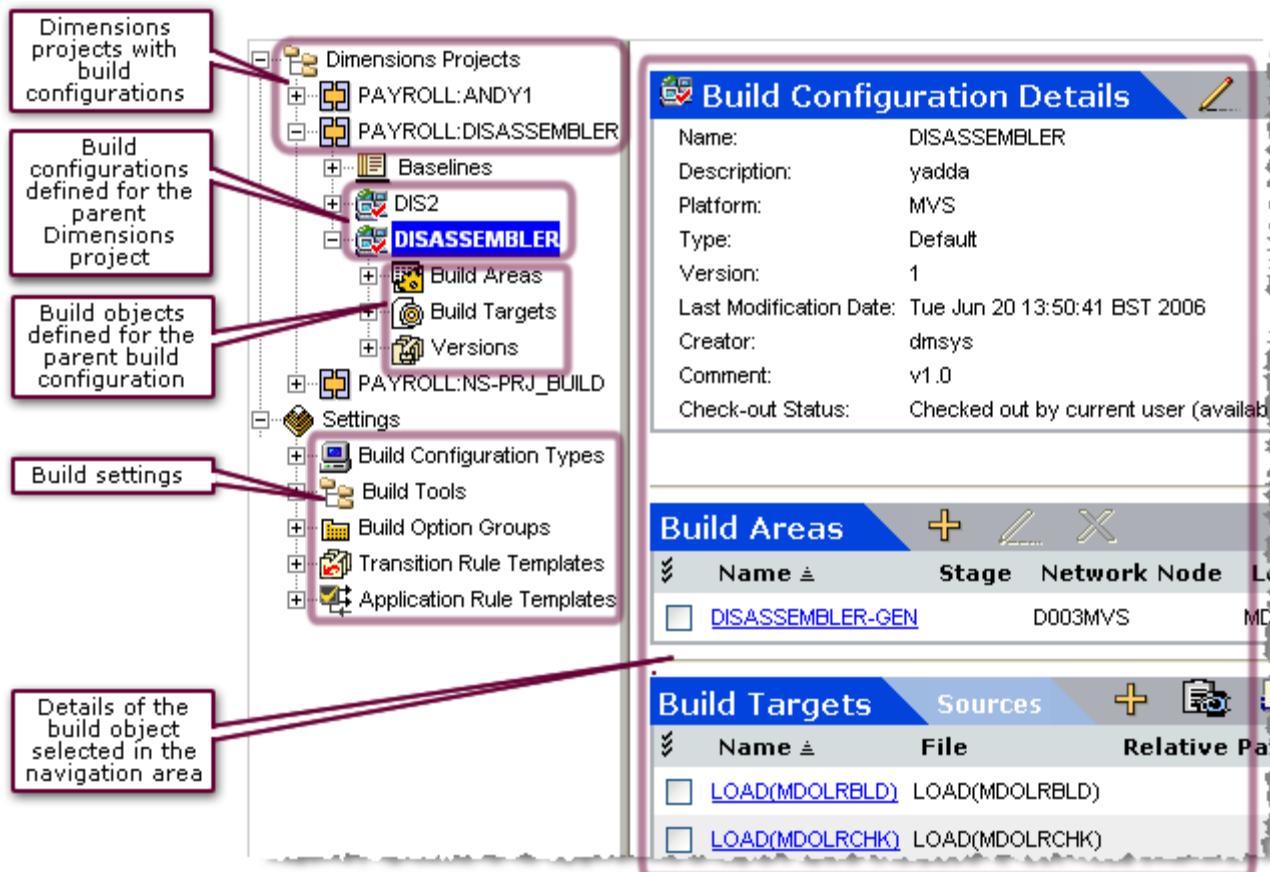
- Support for the Websphere 6.0 application server.
- When you upgrade from Dimensions 9.x all build projects are converted to build configurations.
- The Build Progress display has been improved to show pending messages from the Build Agent that are waiting to be processed.
- The Impacted Targets dialog box and the Targets field in the Run Build dialog boxes in all Dimensions CM clients have been enhanced to display detailed build dependency information.
- Additional template variables have been added to assist build template writers.

Dimensions Build Terminology

Build area	The file system location used during a build to look for inputs and generate outputs. A build area definition includes the name of the logical or physical node where the build area is located, the port number running the Dimensions agent, the file system location, and the user credentials used to secure access to the build area. Build areas can be of two types, <i>work areas</i> and <i>deployment areas</i> . For details see page 133 .
Build configuration	Contains information about what will be built and how, including the targets, build areas, options, scripts, and dependencies for a specific platform or environment. For details see page 183 .
Build configuration type	The options that are required for a particular type of build configuration. For details see page 152 .

Global Stage Lifecycle	The Global Stage Lifecycle is the lifecycle that items follow that controls which versions are included in the configurations and builds of a project. Item revisions are moved to the next stage in this lifecycle when they have reached the appropriate stage of approval (a process called <i>deployment</i>). This lifecycle is defined for the base database. If any deployment areas are associated with these stages the item files are copied to those areas when the items are deployed to the corresponding stage. For details about editing the Global Stage Lifecycle see the <i>Process Modeling User's Guide</i> .
Build tool	The logical groupings of build options for a given build engine. For example, you can configure all the possible options for the Microsoft Visual C++ compiler and add a description for each option. For details see page 157 .
Build option group	A logical collection of build tool options that are passed to the build engines and scripts that execute builds. Enables you to apply rules and enforce standards. For details see page 163 .
Template	A customizable text file containing variables and control words. Dimensions supports a common templating language and processor across Windows, UNIX, Linux, and z/OS (both USS and MVS). The templating language enables you to customize the processing of build, e-mail, and remote job execution templates on all Dimensions nodes. For details see the <i>Developer's Reference</i> .
Transition rule template	Describes a generic rule for building an item and applying it to specific source items at a later time. It describes how to change or transition an item of one type (i.e., *.c) to an item of another type (for example, *obj) using a template or script. For details see page 169 .
Application rule template	Defines a group of transition rule templates that are used to create an application. For more details see page 179 .
Primary Build Execution Monitor	On mainframe platforms the Primary Batch Execution Monitor (PBEM) controls, monitors, and reports results for builds that are initiated by Dimensions Build. On distributed platforms the PBEM run templates for builds that are initiated by Dimensions Build. For details see page 324 .
The Secondary Build Execution Monitor	The Secondary Build Execution Monitor (SBEM) is an MVS only generalized batch execution tool. For details see page 327 .

The diagram below illustrates some of the features described above.



The Build Flow

The table below lists the high-level flow that you need to follow to set up and run builds. For details about each step see the chapter or document that is referenced.

Step	See this chapter or book
Set up network nodes for all remote build machines.	Administrator's Guide
Set up build areas (deployment and work areas).	Process Modeling User's Guide
(Optional) Edit the Global Stage Lifecycle.	Process Modeling User's Guide
(Optional) Configure preservation policies and rules for the Dimensions item types that will participate in your build.	<ul style="list-style-type: none"> ▪ "Preservation Rules and Policies" on page 137 ▪ Process Modeling User's Guide

Step	See this chapter or book
(Optional) Configure upload rules for the Dimensions item types that will participate in your build.	Process Modeling User's Guide
Check that the item types that you will be using in your build can participate in a build execution.	Process Modeling User's Guide
(Optional) Customize and write build templates.	Developer's Reference
Set up Dimensions projects and add items.	User's Guide
Add deployment areas to Dimensions projects.	User's Guide
(Optional) Configure build settings.	"Managing Build Settings" on page 151
Add build configurations to the Dimensions projects.	"Managing Build Configurations" on page 183
Add build areas to the build configurations.	"Managing Build Configurations" on page 183
Add or import targets to the build configurations. Add sources, scripts, and build options as required.	"Managing Build Configurations" on page 183
Check in the build configurations.	"Managing Build Configurations" on page 183
Test the build configurations and fix any problems that may occur in the builds.	"Executing Builds in the Build Administrator" on page 227
(Optional) Set up notifications.	"Managing Notifications" on page 277
(Optional) Set up build schedules.	"Scheduling Build Jobs" on page 249
Run the production builds from Dimensions clients or Dimensions Build.	<ul style="list-style-type: none"> ■ "Executing Builds in the Build Administrator" on page 227 ■ User's Guide
Monitor the builds.	"Monitoring Builds in the Build Administrator" on page 261

User Scenarios

The following scenarios illustrate typical ways that you can use Dimensions Build to manage and build source code. The scenarios use two fictitious characters, Maya, a developer, and Tammy, a build manager.

Developer Builds

This scenario describes how you can use Dimensions Build to build specific items.

- 1** Tammy creates a Dimensions project for a product, creates the build configurations, and adds the source items to the build configurations.
- 2** Maya uses a Dimensions client to check out the source code that she needs to edit to her work area.
- 3** Maya uses her integrated development environment (IDE) to edit and build her source code. Maya adds new source code as required.
- 4** When Maya is satisfied that her code is ready, she checks the sources back into Dimensions.
- 5** Tammy decides that some of Maya's source code needs to be added to a test environment. Tammy imports Maya's build configuration, build.xml or a makefile, into Dimensions Build.
- 6** Tammy deploys the required items to the next stage in the Global Stage Lifecycle. The code is deployed to the deployment area associated with the build configuration. This process is driven by the Dimensions process model created by the Dimensions administrator.
- 7** Tammy selects the new items that were deployed and initiates a build. Only the targets affected by those items are built.
- 8** Tammy uses the notification feature in Dimensions Build to send e-mail updates to herself and Maya indicating the success or failure of the build.

Project Builds

This scenario describes how, Tammy, who is responsible for all project level builds for the development team, executes project builds using Dimensions Build. When you build a project all targets affected by any item in the project are built.

- 1** Tammy sets up a Dimensions project, a build configuration, and a Unit Test deployment area. She associates this deployment area with the project and the build configuration.
- 2** Tammy uses Dimensions Build to create a schedule that executes a nightly build of all the code that is at the Unit Test stage in the Global Stage Lifecycle. Tammy also configures the build process to capture outputs.
- 3** At the end of every day the developers check in their code and promote it to the Unit Test stage. This deploys the code to the deployment area associated with the build configuration, which is administered by Tammy. This process is driven by the Dimensions process model created by the Dimensions administrator.

- 4 Each morning Tammy uses the build monitoring features of Dimensions Build to check the status of the build that executed overnight and to check for errors. The outputs of the build were captured automatically and added to Dimensions, and are available to the development team.

Continuous Developer Builds

This scenario describes how Tammy uses Dimensions Build to perform continuous builds throughout the working day.

- 1 Tammy uses the Area Definitions cluster of the administration console to create a local work area.
- 2 Tammy uses Dimensions Build to add a build configuration to the Dimensions project and adds the work area to the build configuration.
- 3 Tammy sets up a notification schedule to send e-mail updates to herself to report the success or failure of the builds.
- 4 Tammy sets up a schedule to build the configuration every two hours during the working day.
- 5 Each member of the development team use a Dimensions client to check out the items that they need to edit.
- 6 Each member of the development team uses their preferred development environment to edit their sources, and then checks them back in to Dimensions.
- 7 Every two hours Tammy receives an e-mail notifying her about the status of the most recent build.

Baseline Builds

This scenario describes how Tammy creates baseline builds and builds to a clean work area. When you build a baseline you build all the targets that are affected by any items in the baseline; the build configuration and sources used in the build are the versions at the time the baseline was taken. When you create baselines for a project that is associated with a build configuration, the baselines appear in the build configuration in Dimensions Build.

- 1 Tammy sets up a work area for the baseline and associates it with the project and build configuration she created earlier.
- 2 The product that Tammy is working on is ready to go to beta testing. Tammy uses the desktop or web clients to create a baseline that captures the current state of the build configurations and sources in the project.
- 3 At some later date Tammy has to patch a bug in the version of the product that was tested and delivered to customers. To build the product exactly as it was when the baseline was taken, Tammy selects the correct baseline in Dimensions Build and initiates a build. A baseline in Dimensions Build lists all the inputs, outputs, options, and scripts at the time the baseline was taken.
- 4 After the product is rebuilt Tammy adds the source code that was fixed and sends the product for testing.

Request Builds

This scenario describes how the build manager Tammy builds a request. When you build a request all the targets affected by items that are related *In Response* to the request are built.

- 1 A software application has been built and is running in a live deployment area that is used on a continuous basis by customers. A critical problem is found in the software that needs to be urgently fixed.
- 2 Tammy creates a new request for the problem. She then investigates the source of the problem and relates the affected source items to the request.
- 3 Tammy delegates the problem to Maya. When making the changes Maya finds a related problem in another item and fixes it using a new request. She then relates the second request to the first request using a child relationship.
- 4 Tammy uses the desktop or web clients to build the request, taking care to select the correct build configuration and deployment area. She also includes items from related child requests.
- 5 After the build has executed successfully Tammy deploys the request to the live deployment area.

Build Areas

A build area is the physical location where a build takes place. A build area can be a Dimensions work area or a deployment area.

- A *work area* is an area on your hard drive or a remote node where Dimensions CM performs all file operations such as check in and check out. All operations are relative to the folder that is referenced by your work area or project root folder. A work area can also be shared by a group of users.
- A *deployment area* contains items that have reached a particular stage of development in the Global Stage Lifecycle (GSL). When items are deployed to a stage in the GSL you can also have them automatically moved to the deployment areas associated with that stage. For more information about deployment see the *Introduction to Dimensions CM* and the *User's Guide*.

Build Area Locations

You can associate one or more build areas with each build configuration. You can locate build areas on:

- Your current machine.
- A Dimensions application server machine.
- A remote node, using the normal Dimensions remote node identification mechanism.

A Dimensions agent is required on each machine hosting a build area, for details see the *Installation Guide*.

On a mainframe machine you can have multiple logical MVS and USS nodes, each running a Dimensions agent. For information about setting up mainframe physical and logical nodes see the *Dimensions for z/OS User's and Administrator's Guide*.

For information about setting up physical and logical nodes all other platforms see the *Process Modeling User's Guide*.

Build Area Security and Ownership

When you setup a build area you must specify a Dimensions user ID and password to be able to connect to the area during a build. Dimensions Build does not use the credentials of the current logged in Dimensions user in case these are not available during a build, for example, when a scheduled build runs at night. You can also specify that the build area password is requested when a build is launched. If the user ID cannot be authenticated on a build area, no request is made for a password and the build does not start.

Each build area that you set up can have a different owner and be on a different machine. However, all the build areas used in a Dimensions project must be on the same network node, or accessible from their network nodes via NFS shares (on UNIX) or network drives (on Windows).

Build operations are performed in the security context of the user who owns the build area. For example, if your UNIT TEST build area is owned by Jim, items are written to that area in the security context of user Jim, regardless of which user name you used to log in to Dimensions. The builds also run as the build area owner, Jim. However the `-lo` parameter remains set to the Dimensions user name you used to log in, and enables you to view the Dimensions Build logs.

The user name submitted to Dimensions Build via the `-lo` log owner parameter is always the name of the user who logged into Dimensions.



NOTE

- When you create a build area you must ensure that the build area owner has permission to read and write to any files in the directory.
- If you do not specify a build area owner the build area is populated using the credentials of the current user. Serena recommends that you always specify a build area owner.

Build area user credentials are determined by Dimensions in the following order:

- 1 The credentials of the build area owner.
- 2 The user credentials specified with the last AUTH command for the remote node.
- 3 The current user's log in credentials. If the user logged in using LDAP credentials, they will probably not be valid operating system user credentials.
- 4 If none of the above credentials authenticate, you will be prompted with a log in dialog box to enter credentials for the build area node.

Builds at the DEVELOPMENT stage in the current project root directory are submitted and performed in the security context of the user running the Dimensions client. If your project root directory contains a node name, normal Dimensions security applies; the client tries your Dimensions credentials and if authentication fails, prompts you for the correct user credentials.

For more information about build security see [Appendix D, "Dimensions Build Security" on page 339](#).

Managed Development Areas

Managed development areas are deployment areas that are assigned to the first stage in the global stage lifecycle, typically the DEVELOPMENT stage. You can use managed development build areas as integration build areas for source code changes done by a development team when the code is not ready for deployment to a stage such as UNIT TEST or SYSTEM TEST.

Build areas are maintained the same as all other areas. For example, if you create an item revision it is automatically exported to all managed development areas associated with the first stage. If you deploy an item revision from the DEVELOPMENT to UNIT TEST stages, it is removed from all managed DEVELOPMENT areas.

Build Areas and Active Item Revisions

The areas associated with a stage in the global stage lifecycle only contain active item revisions. An item revision at a stage is only active when there are no newer revisions of the same item at that stage, or any later stage.

- If the item revision becomes the active revision at the new stage, a copy of the item revision is automatically placed in the area(s) mapped to the new stage.
- If the item revision was the active revision at the old stage, the item revision is deleted from the area(s) mapped to the old stage.

For example, if you deploy foo.c from UNIT TEST to SYSTEM TEST:

- A copy of foo.c is placed in the areas associated with the SYSTEM TEST stage.
- foo.c is deleted from the areas associated with the UNIT TEST stage.

Search Paths

If you are using a build engine that supports the concept of search paths, such as Openmake or the Serena mainframe build engine, to set up search paths in Dimensions Build do the following:

- 1 Attach your deployment areas to your Dimensions project (use the desktop client; for details see the *User's Guide*).
- 2 Attach your deployment areas to your build configuration, see ["Managing Build Configurations" on page 183](#).
- 3 Set the DMPATH variable in your build template to list the search paths. The Openmake and mainframe templates supplied with Dimensions will honor those search paths.

For details about integrating with Openmake see [page 301](#).

For details about the Serena mainframe build utility see [page 323](#).

For details about build templates see the *Developer's Reference*.

Automatic Builds

Dimensions enables you to automatically start a build when the deployment of an item is completed. In the Deploy Item dialog box select the *Start build after deployment* option. If you select this option, the targets at a specific stage are always up to date. For details see the *User's Guide*.

Builds will only execute automatically when:

- The deploy command completes successfully.
- The item type can participate in a build execution.
- One or more build targets are impacted by the item revision you selected.

Build Roles and Privileges

The table below lists the Dimensions privileges that are required to administer and launch builds.

Build Function	Privilege Required
Build administration	Manage build configurations
Build items Build projects	Build from a project
Build requests	Build from a project
Build baselines	Build from a baseline

For more information about privileges and how to define them see the *Process Modeling User's Guide* or the Administration Console help.

Capturing Build Outputs

Dimensions enables you to capture build outputs and check them into Dimensions. This functionality is also referred to as a closed-loop build. If you choose to capture build outputs the following actions occur after the execution of a build job is completed:

- Dimensions Build captures the build targets defined in the build configuration.
- By default, build targets are checked into the Dimensions project from where you initiated the build. You can choose to put the targets in a different project. The default Dimensions upload rules automatically derive the file format and item type of the outputs (see the *Process Modeling User's Guide* for details).

The upload rules for MVS build outputs stored in a PDS/E or PDS, such as MERVK.VT.LOAD(SUBB), are defined in terms of the data set low level qualifier, converted to lower case, for example, %load. By default, listing data sets are held per module. The upload rules for generated listings, such as MAL.UT.SUBB.LLISTINGS, are also defined in terms of the data set low level qualifier, converted to lower case, for example, %llisting. Listing data sets can also be placed in libraries

(PDS/E or PDS) for which the upload rules are defined the same as for other build outputs stored in a PDS/E or PDS.

- Preservation rules specify whether to preserve each build output as a normal, placeholder, or external item revision. See below for details.
- The build outputs are created in Dimensions at the initial lifecycle state and the initial stage, for example, Development.
- The Dimensions server creates relationships between every captured target and the source items used to build the target. The server also creates a post-build bill of materials report showing all the components (revisions) that went into the build. This report is stored in Dimensions and any e-mail notifications that have been set up are sent to subscribers.
- The Dimensions server relates the captured outputs to the requests that you optionally specified when you initiated the build.
- If the build used inputs that are not under the control of Dimensions, this information is stored in the Dimensions repository and in the made-of report. For example, if you are using Openmake the search path includes locations that are not defined as Dimensions build areas.

For details about capturing build outputs and viewing items derived from closed-loop builds see the *User's Guide* or the online help for the desktop and web clients.

Preservation Rules and Policies

If you choose to preserve build outputs each built target is mapped by upload rules to a particular Dimensions item type. The rules in a preservation policy define whether each built target is preserved as a normal, external, or placeholder item revision. By default, all built targets are preserved as normal Dimensions item revisions.

External Item Revisions

External item revisions represent versioned files whose actual content is stored outside of a Dimensions item library. The location of an external revision, which comprises a network node name and a full path to a file on that network node, is stored in a Dimensions base database. Typically, you use external item revisions to represent compile and link listings generated as a result of a build on a z/OS mainframe node.

Physically, an external item revision is represented as a zero-byte file in the item library. You can only create an external item revision for a build output, such as listing, that was generated outside a build area. Each external item revision must represent a unique file.

You can deploy external item revisions between stages. However, the file referred to by the external file revision is not deployed to build areas. If you rename, move, or delete the external file revision the build area is not updated. The AUDIT command ignores external item revisions.

You can fetch and extract external item revisions. If you use a Dimensions client to revise an external item revision, by executing RI or UI commands, a normal item revision is created.

Placeholder Item Revisions

Placeholder item revisions represent versioned files that have no content in a Dimensions item library; a placeholder item revision has no storage at all. Typically, you use placeholder item revisions to represent intermediate targets and made-of relationships, avoiding the overhead in Dimensions of preserving every build output generated as a result of a build job.

Physically, a placeholder item revision is represented as a zero-byte file in the item library, and can be created for any build output inside or outside a build area. You can deploy placeholder item revisions between stages. However, since a placeholder item revision does not refer to any files, no build areas are updated; instead, deployment is reduced to a stage change. The AUDIT command ignores placeholder item revisions.

You cannot fetch or extract placeholder item revisions. If you use the UI command in a Dimensions client to revise a placeholder item revision, a normal item revision is created. You can only create placeholder item revisions from build output collections.

For details about creating preservation rules and policies see the *Process Modeling User's Guide*.

Auditing Build Areas

The AUDIT command produces an audit report for the build areas associated with a Dimensions project. The report compares the files in the Dimensions repository with those in the build areas that you select. You can also repair the build area so that it contains all the item revisions from the project that are at the build stage you selected.

For information about auditing build areas see the *User's Guide* or the online help for the desktop and web clients. For information about the AUDIT command see the *Command-Line Reference*.

Templates

A template is a customizable text file containing variables and control words. In Dimensions, templates are used to execute builds and execute general commands via remote job execution. You can also attach scripts and templates to deployment areas. For details see the *Developer's Reference*.

In addition, you can use templates to construct the body of e-mail messages used for notifications, for details see the *Process Modeling User's Guide*.

Impacted Targets

Dimensions enables you to view the targets impacted by items, a request, or a project. This is useful when you want to check what targets will be built before you launch a build or what targets will be affected if you change a source file. For details see the *User's Guide* or the online help for the desktop and web clients.

Chapter 5

Overview of the Build Administration User Interface

About Build Administration	140
Build Administration Main Window	140
Build Management Tab	141
Build Scheduling Tab	143
Build Monitoring Tab	145
Notification Tab	146

About Build Administration

Dimensions Build is a build management, execution, and monitoring tool that is part of the Dimensions CM administration console. For an overview of Dimensions Build see [page 125](#).

Invocation Dimensions Administration Console | Distributed Development | Build Administration

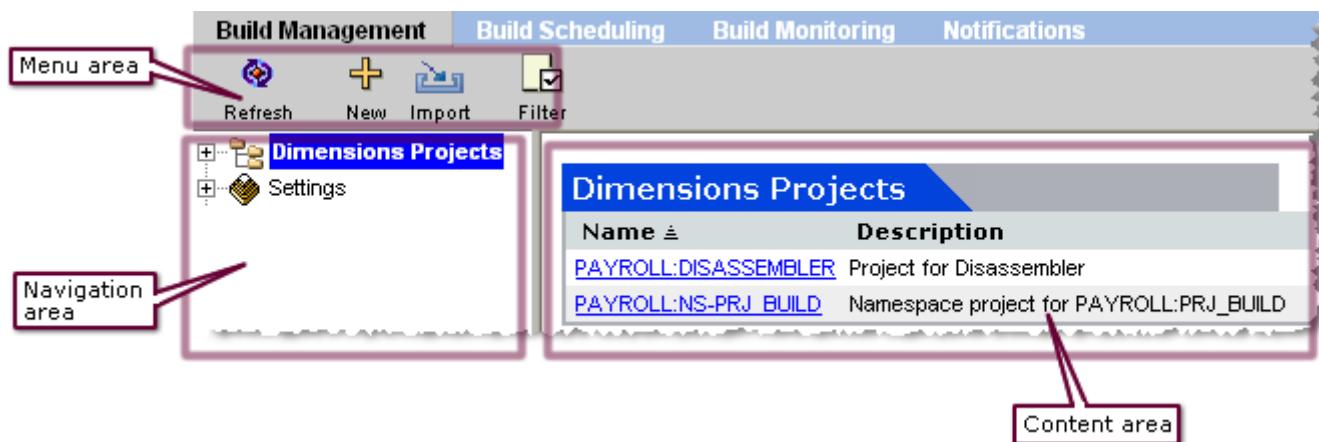
Build Administration Main Window

The Build Administration main window has the following tabs:

- **Build Management:** Enables you to manage build configurations and settings, see [page 141](#).
- **Build Scheduling:** Enables you to execute builds and set up build schedules, see [page 143](#).
- **Build Monitoring:** Enables you to monitor the progress of your builds and view the history of previous builds, see [page 145](#).
- **Notifications:** Enables you setup e-mails that notify you about the progress of your builds, see [page 146](#).

Each tab has the following areas:

- **Menu area:** Displays a toolbar to help you carry out various build tasks. The buttons that are displayed depend on the object that you select in the navigation or content areas.
- **Navigation area:** Allows you to view and select build objects.
- **Content area:** Displays details about the build object that you have selected in the navigation area.



Build Management Tab

The Build Management tab has the following objects:

The screenshot illustrates the Build Management Tab interface. On the left, a navigation tree titled 'Dimensions Projects' shows a hierarchy of build configurations. A callout box labeled 'Dimensions projects with build configurations' points to the root node. Another callout box labeled 'Build configurations defined for the parent Dimensions project' points to the 'DISASSEMBLER' node under PAYROLL:DISASSEMBLER. A third callout box labeled 'Build objects defined for the parent build configuration' points to the 'Build Areas', 'Build Targets', and 'Versions' nodes under DISASSEMBLER. A fourth callout box labeled 'Build settings' points to the 'Build Configuration Types', 'Build Tools', 'Build Option Groups', 'Transition Rule Templates', and 'Application Rule Templates' nodes under Settings. A fifth callout box labeled 'Details of the build object selected in the navigation area' points to the 'DISASSEMBLER' node in the navigation tree.

Build Configuration Details

Name:	DISASSEMBLER
Description:	yadda
Platform:	MVS
Type:	Default
Version:	1
Last Modification Date:	Tue Jun 20 13:50:41 BST 2006
Creator:	dmsys
Comment:	v1.0
Check-out Status:	Checked out by current user (available)

Build Areas

Name	Stage	Network Node
DISASSEMBLER-GEN	D003MVS	MD

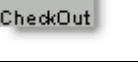
Build Targets

Name	File	Relative Path
LOAD(MDOLRBLD)	LOAD(MDOLRBLD)	
LOAD(MDOLRCHK)	LOAD(MDOLRCHK)	

Dimensions Projects

When you are working in the Dimensions Projects section of the Build Management tab, the menu area has the following buttons. The buttons that are displayed depend on the object that you select in the navigation or content areas.

Button	Description
	Refresh: refreshes the screen.
	New: creates one of the following new objects: <ul style="list-style-type: none"> ■ Build area. ■ Build target.
	New: creates a new build configuration.

Button	Description
 Import	Import: imports a build configuration from an XML file.
 Edit / View	Edit/View: displays the selected build configuration and enables you to edit it.
 Edit	Edit: edits the object selected in the content area.
 Delete	Delete: deletes the object selected in the content area.
 Filter	Filter: (when you select the Dimensions Projects node in the navigation tree) filters the Dimensions projects that are displayed in the navigation and content areas.
 Copy	Copy: copies the selected build configuration or build target.
 CheckIn	Checkin: checks in the selected build configuration.
 CheckOut	Checkout: checks out the selected build configuration.
 UndoCheckout	Undo Checkout: undoes the check out of the selected build configuration.
 Run	Run: builds target(s) in the selected build configuration.
 View	View: displays details of the selected checked in build configuration.
 Rollback	Rollback: rolls back your current build configuration to the version you selected in the Versions node of the navigation area.
 Rewrite	Rewrite: overwrites your current build configuration with the version you selected in the Versions node of the navigation area.

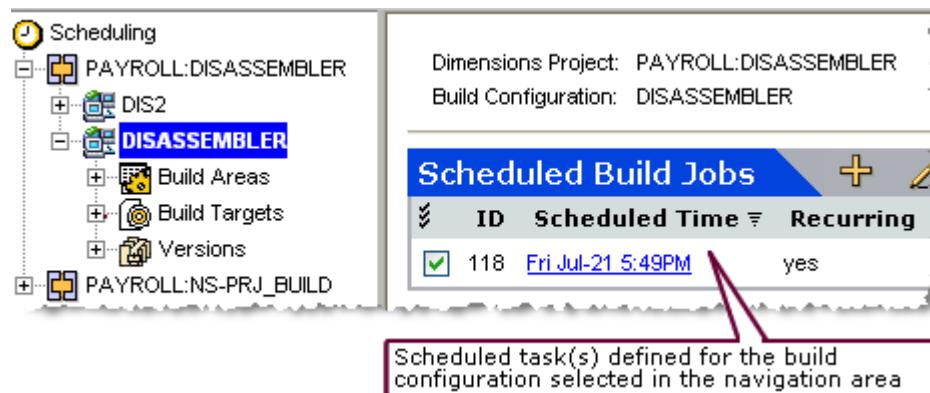
Settings

When you are working in the Settings section of the Build Management tab, the menu area has the following buttons. The buttons that are displayed depend on the object that you select in the navigation or content areas.

Button	Description
 Refresh	Refresh: refreshes the screen.
 Import	Import: (when you select the Build Tools node in the navigation tree) imports a build tool from an external file.
 New	New: creates one of the following new objects: <ul style="list-style-type: none">■ Build configuration type.■ Build tool.■ Build option group.■ Transition rule template.■ Application rule template.
 Edit	Edit: edits the object selected in the navigation or content areas.
 Delete	Delete: deletes the object selected in the content area.

Build Scheduling Tab

The Build Scheduling tab has the following objects:



The screenshot shows the Build Scheduling Tab interface. On the left, there is a navigation tree with nodes like 'Scheduling', 'PAYROLL:DISASSEMBLER' (which is expanded), 'DIS2', 'DISASSEMBLER' (selected and highlighted in blue), 'Build Areas', 'Build Targets', and 'Versions'. Below this is another node 'PAYROLL:NS-PRJ_BUILD'. To the right of the tree, there is a summary box with 'Dimensions Project: PAYROLL:DISASSEMBLER' and 'Build Configuration: DISASSEMBLER'. Below this is a table titled 'Scheduled Build Jobs' with columns 'ID', 'Scheduled Time', and 'Recurring'. One row is shown: '118' with 'Fri Jul-21 5:49PM' under 'Scheduled Time' and 'yes' under 'Recurring'. A red callout box points to this row with the text 'Scheduled task(s) defined for the build configuration selected in the navigation area'.

The menu area has the following buttons. The buttons that are displayed depend on the object that you select in the navigation or content areas

Button	Description
 Refresh	Refresh: refreshes the screen.
 Schedule	Schedule: creates a new build execution schedule.
 Scheduler	Scheduler Service: creates or modifies a scheduler service.
 Filter	Filter: (when you select the Scheduling node in the navigation tree) filters the Dimensions projects that are displayed in the navigation and content areas.
 Run	Run: (when you select a build configuration in the navigation tree) enables you to build the selected build object.

Build Monitoring Tab

The Build Monitoring tab has the following objects:

The screenshot shows the Build Monitoring Tab interface. On the left is a navigation tree with two main sections: "Build jobs that are currently executing" and "History". The "History" section is expanded, showing a tree structure for "PAYROLL:DISASSEMBLER" and "DISASSEMBLER", with a node "Version #1 [Wed Jul-05 10:47A]" selected. A callout box labeled "Details of the build job selected in the navigation area" points to this node. To the right of the navigation tree is a content area titled "Build Job Details" which displays various job details. Below this is another content area titled "Build Monitor Events" which lists five events with icons and descriptions.

Order	Step	Type
1		Build started
2		Task Build started
3		Task Build started
4		Task Build finished
5		Task Build started

The menu area has the following buttons. The buttons that are displayed depend on the object that you select in the navigation or content areas.

Button	Description
	Refresh: refreshes the screen.
	Cancel: (for the Running Jobs node only) cancels the build job that is currently executing.
	Rebuild: rebuilds the build job that you select in the navigation or content areas.
	Filter: filters the build jobs that are displayed in the navigation and content areas.

Notification Tab

The Notification tab has the following objects:

The screenshot shows the Notification tab interface. On the left is a navigation tree with the following structure:

- Notifications
 - Templates
 - Build Completed
 - Build Started**
 - Subscriptions
 - PAYROLL:DISASSEMBLER
 - DIS2
 - DISASSEMBLER
 - Events
 - 22-Build is started()**
 - 23-Build is finished()
 - PAYROLL:NS-PRJ_BUILD

Annotations on the left side of the tree:

 - Notification templates**: Points to the Templates node.
 - Notifications events defined for the parent build configuration**: Points to the PAYROLL:DISASSEMBLER node.

The right side contains two content panes:

 - Notification Event Info**: Displays details for the selected event (Build Started). It includes fields for Project (PAYROLL:DISASS), Build Config (DISASSEMBLER), Target, Event ID (22), Event Type (Build is started), Subject, Message, and Template (Build Started).
 - Notification Subscribers**: Shows a list of subscribers for the selected event. It includes columns for Subscriber (DMSYS), Type (User), and Subject (<%EVENT_TIME%> <%>).

Annotations on the right side of the panes:

 - Details of the notification template or event selected in the navigation area**: Points to the event details in the 'Event Info' pane.
 - Details of notifications that are subscribed to the selected notification event**: Points to the subscriber list in the 'Subscribers' pane.

The menu area has the following buttons. The buttons that are displayed depend on the object that you select in the navigation or content areas.

Button	Description
	Refresh : refreshes the screen.
	Notification : creates or modifies a notification service.
	New : creates a new template or notification event.
	Copy : copies the selected notification template.
	Filter : filters the build jobs that are displayed in the navigation and content areas.

Part 3

Configuring and Managing Builds

Part 3: Configuring and Managing Builds contains the following chapters

Configuring an MVS Build Environment	149
Managing Build Settings	151
Managing Build Configurations	183
Executing Builds in the Build Administrator	227
Using Filters	235
Versioning	243
Scheduling Build Jobs	249
Monitoring Builds in the Build Administrator	261
Managing Notifications	277

Chapter 6

Configuring an MVS Build Environment

Pre-Requisites	150
Setting up your MVS Build Environment	150

Pre-Requisites

Use the Dimensions Administration Console to create definitions for the following remote z/OS mainframe network nodes:

- A physical network node for the z/OS mainframe where the Dimensions listener resides.
- A logical network node and connection for the USS file system.
- A logical network node and connection for the MVS file system.

For details see the *Dimensions for z/OS User's and Administrator's Guide*

Before setting up and running builds, check that you have access to all remote nodes where you will be running builds.

Setting up your MVS Build Environment

Setting up builds on z/OS mainframes can be a complicated and lengthy process. Serena recommends that you follow the steps below to set up your z/OS build environment:

1 Check that the following software is installed and configured:

- A Dimensions server on any supported platform. For details see the *Installation Guide*.
- Dimensions for z/OS: check that you can run the Dimensions for z/OS ISPF client, or perform a remote checkout from the desktop client to the Dimensions node. For details see the *Dimensions for z/OS User's and Administrator's Guide*.

2 On your z/OS mainframe do the following:

- Pre-allocate the build areas that Dimensions will use to ensure that data sets are allocated properly (DCB, sizes, etc.) for your application and project. As a general rule, the last qualifier in the data set name indicates its type, for example, COBOL, COPY, H, and C. Build areas are typically allocated for each stage. Optionally, you can allocate a managed Development build area.



NOTE Any managed build areas that you set up should be allocated under a security protected ID that the average Dimensions user does not have write access to. This precaution is necessary to protect the integrity of the data contained in the build areas.

- Create data sets for targets and intermediate targets, for example, .LOAD and .OBJ respectively.
- (Optional) Set up a user build area where you will do development work; allocate a collection of data sets with a common HLQ ('directory') as required for your build.

3 (Optional) If you are adding items from a PC to a z/OS mainframe and want automatic allocation of data sets, check that the Dimensions configuration file on the z/OS mainframe, MDH.V1010.MDHPARM(MDHTDCFG), has the correct defaults to create data sets by LLQ (low level qualifier). For details see the *Dimensions for z/OS User's and Administrator's Guide*. Serena recommends that you perform step 2 above to ensure proper dataset sizing for each item type.

Chapter 7

Managing Build Settings

About Build Settings	152
Build Configuration Types	152
Build Tools	157
Build Option Groups	163
Transition Rule Templates	169
Application Rule Templates	179

About Build Settings

Dimensions Build enables you to configure the following build settings:

- **Build Configuration Types**

Build configuration types specify options that are required for a particular type of build configuration. For more detail see the next section.

- **Build Tools**

Build tools are logical groupings of build options for a given build engine. For more details see [page 157](#).

- **Build Option Groups**

Build option groups are logical collections of build tool options that are passed to the build engines and scripts that execute builds. For more details see [page 163](#).

- **Transition Rule Templates**

Transition rule templates describe generic rules for building items. For more details see [page 169](#).

- **Application Rule Templates**

Application rule templates define groups of transition rule templates that are used to create applications. For more detail see [page 179](#).

Build Configuration Types

About Build Configuration Types

Build configuration types enable you to specify options that are required for a particular type of build configuration. For example, for the Openmake build configuration type you must specify the URL to the Openmake build monitor as follows:

Option: DMOMSERVER (Openmake server path)

Value: `http://localhost:58080`



TIP When you add a build configuration to a Dimensions project you have to specify a build configuration type. To specify a build configuration type that is not one of the Dimensions Build default types, you must add it first.

Dimensions Build includes the following build configuration types:

- Openmake: for Openmake builds.
- Default: for all other builds.



NOTE You cannot modify or delete the base properties of the default build configuration types, or add new options. However you can modify the default options. To add new options you can add build options to your build configuration, for details see "[Build Options](#)" on page 219.

About Build Configuration Type Options

Build configuration type options have the following properties:

- **Name:** the name of the option.
- **Description:** the description of the option.
- **Level:** must be one of the following:
 - **Build Configuration:** an option that is specific to the entire build configuration type, for example, Openmake project name or Openmake search path.
 - **Build Area:** an option that is specific to a build area, for example, a path to license file, a license manager host, or a Dimensions root directory.
 - **Build Transition:** an option that is specific to a script that builds targets, for example, debug or trace.
- **Default Value:** the default value of the option.
- **Required:** specifies if the option is mandatory.

About the Openmake Build Configuration Type

The Openmake build configuration type that is included with Dimensions Build has the following default options and values:

Option Name	Description	Level	Default Value	Required
DMBLDPROJ	Openmake project name	Build configuration	None	Yes
DMOMINSTALL	Openmake installation path	Build Area		Yes
DMOMSERVER	Openmake server path	Build Area	http://localhost:58080	Yes
DMPUBLIC	Public build job	Build configuration	True	Yes
DMSEARCHPATH	Openmake search path	Build configuration		Yes
DM_LICENSE	Path to the license file or license manager host.	Build area	@localhost	No
DM_OM_OSPLATFORM	Openmake platform	Build configuration	Java	Yes

The variables listed above are inputs to Openmake build templates. For details about the templating language and the Openmake build templates see the *Developer's Reference*.

Viewing Build Configuration Types

Purpose Follow this procedure to view a list of all the build configuration types currently defined.

To view build configuration types:

In the navigation area of the Build Management tab click Settings and click Build Configuration Types. The content area lists all the build configuration types that are currently defined.

Adding Build Configuration Types

Purpose Follow this procedure to add a new build configuration type.

To add a build configuration type:

- 1 In the navigation area of the Build Management tab click Settings and click Build Configuration Types.
- 2 In the content area click New Object.



The Add New Build Configuration Type dialog box appears.

- 3 For **Name** type the name of the build configuration type.
- 4 For **Code** type a description of the build configuration type.
- 5 Click OK. The new build configuration type is added to the list of build configuration types in the content area.

Adding Options to Build Configuration Types

Purpose Follow this procedure to add a new option to a build configuration type.



NOTE You cannot add new options to the default build configuration types.

To add an option to a build configuration type:

- 1 In the navigation area of the Build Management tab click Settings, click Build Configuration Types, and click the build configuration type where you want to add an option. The content area refreshes and displays details of the build configuration type.
- 2 In the content area click New Object.



The Add New Build Configuration Type dialog box appears.

- 3 For **Name** type the name of the build configuration type option.

- 4 For **Description** type a description of the build configuration type option.
- 5 From the **Level** list select the level that is appropriate to the option.
- 6 For **Default Value** type a default value for the option.
- 7 To make this option mandatory, select the **Required** check box.
- 8 Click OK. The new option is added to the list of build configuration type options in the content area.

Modifying Build Configuration Types

Purpose Follow this procedure to modify an existing build configuration type.



NOTE You cannot modify the properties of the default build configuration types.

To modify a build configuration type:

- 1 In the navigation area of the Build Management tab click Settings and click Build Configuration Types.
- 2 In the content area select the check box next to the build configuration type that you want to modify and click Edit Object.



The Edit Build Configuration Type dialog box appears.

- 3 Modify the name if required.
- 4 Modify the Code (description) if required.
- 5 Click OK.

Modifying Options for Build Configuration Types

Purpose Follow this procedure to modify the existing options for a build configuration type.

To modify the options for a build configuration type:

- 1 In the navigation area of the Build Management tab click Settings and click Build Configuration Types.
- 2 In the navigation or content areas select the build configuration type containing the option that you want to modify. The content area refreshes and displays the options defined for the build configuration type.

- 3 Select the check box next to the option that you want to modify and click Edit Selected Object.



The Edit Build Configuration Type Option dialog box appears.

- 4 For **Name** modify the name of the build configuration type option.
- 5 For **Description** modify the description of the build configuration type option.
- 6 From the **Level** list select the level that is appropriate to the option.
- 7 For **Default Value** modify the default value for the option.
- 8 To make this option mandatory, select the **Required** check box.
- 9 Click OK.

Deleting Build Configuration Types

Purpose Follow this procedure to delete build configuration types.



NOTE You cannot delete the default build configuration types.

To delete build configuration types:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the content areas select the check box next to each build configuration type that you want to delete.
- 3 Click Delete. To confirm that you want to delete the build configuration types click Yes.

Deleting Options from Build Configuration Types

Purpose Follow this procedure to delete options from a build configuration type.



NOTE You cannot delete the default build configuration type options.

To delete options from a build configuration type:

- 1 In the navigation area of the Build Management tab click Settings and click Build Configuration Types.
- 2 In the content or navigation areas select the build configuration type containing the options that you want to delete.
- 3 In the Build Configuration Type Options section of the content area select the check box next to each option that you want to delete.
- 4 Click Delete. To confirm that you want to delete the options click Yes.

Build Tools

About Build Tools

Build tools are logical groupings of build options for a given build engine. For example, you can configure all the possible options for the Microsoft Visual C++ compiler and add a description for each option. When you set up a build configuration you can then choose from a meaningful list of possible build options for the compiler. This helps to avoid mistakes when manually entering options each time, and eliminates the need to check the meaning of option syntax in the compiler reference guides.

Build tools also enable you to enforce standards across all builds by managing the build options supported by the build tools used in build scripts.

You can import a build tool from an external XML files, for example:

```
<?xml version="1.0" encoding="UTF-8"?>
<BuildTools author="Mickey Mouse" comments="yadda yadda" >
    <BuildTool name="C++ Compiler" description="C++ Compiler tool">
        <Option name="debug" description="blah blah">
            <Value description="option value description">-debug:1</Value>
            <Value description="option value description">-debug:2</Value>
            <Value description="option value description">-debug:3</Value>
            <Value description="option value description">-debug:4</Value>
        </Option>
        <Option name="#define" description="example of the option without
values - freeform option value"/>
    </BuildTool>
    <BuildTool name="C++ Linker" description="C++ Linker tool">
        <Option name="debug" description="some option description here">
            <Value description="option value description">-debug:1</Value>
            <Value description="option value description">-debug:2</Value>
            <Value description="option value description">-debug:3</Value>
            <Value description="option value description">-debug:4</Value>
        </Option>
    </BuildTool>
</BuildTools>
```

About Build Options

For details of the build options that you can add to build tools, see *Dimensions Build Predefined Symbols* in *The Templating Language and Processor* chapter of the *Developer's Reference*.

Build Tool Examples

- IBM COBOL compiler v1.4

Build Option	Description	Possible Values and Descriptions
INTDATE	Specifies the starting date for integer dates used with date intrinsic functions.	INTDATE(ANSI): Uses the Standard COBOL 85 starting date. Day 1 is Jan 1, 1601. INTDATE(LILIAN): Uses the Language Environment Lilian starting date. Day 1 is Oct 15, 1582
MAP	Produces a listing of the items defined in the DATA DIVISION.	MAP: produces a listing. NOMAP: does not produce a listing.

- Microsoft C++ Compiler

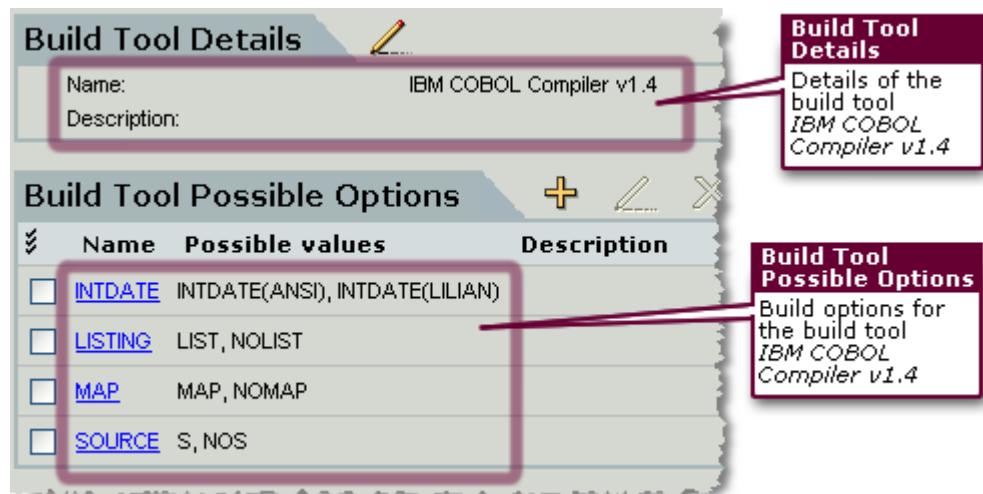
Build Option	Description	Possible Value
INCLUDE	Specifies an INCLUDE directory.	C:\Program Files\Microsoft Visual Studio .NET 2003\VC7\include
LINK	Passes one or more linker options to the linker.	/COMMENT:"For Payroll Project" / DEBUG

Viewing Build Tools and Options

Purpose Follow this procedure to view a list of build tools and their options.

To view build tools and options:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools. The content area lists all the build tools that are currently defined.
- 2 To view a build tool's options, in the navigation or content area click the build tool. The content area refreshes and lists all the options defined for the build tool.



Adding Build Tools

Purpose Follow this procedure to add a new build tool.

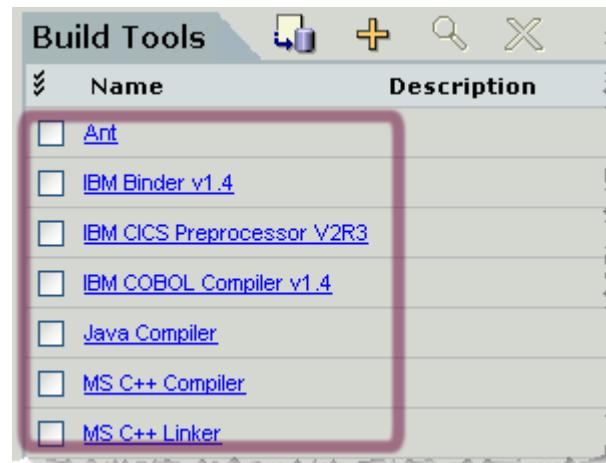
To add a build tool:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the content area click New Object.



The Add New Build Tool dialog box appears.

- 3 For **Name** type the name of the build tool.
- 4 For **Description** optionally type a description of the build tool.
- 5 Click OK. The new build tool is added to the list of build tools in the content area.



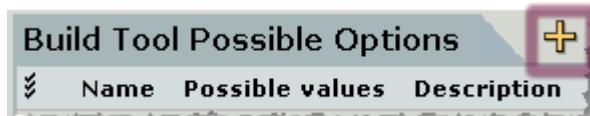
Adding Options to Build Tools

Purpose Follow this procedure to add a new option to a build tool.

To add an option to a build tool:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the content or navigation areas click the build tool where you want to add a build option. The content area refreshes and displays the details of the build tool.

- 3 On the Build Tool Possible Options tab click New Object.



The Add New Build Tool Option dialog box appears.

- 4 For **Option Name** type a name for the option.
- 5 For **Option Description** type a description of the option.
- 6 For **Possible Values** type a value for the option. Press the Tab key.
- 7 For **Descriptions of Values** optionally type a description of the value.
- 8 To add additional values to this option repeat steps 6 and 7.
- 9 Click OK. The build options and values are added to the Build Tool Possible Options tab in the content area.

	Name	Possible values	Description
<input type="checkbox"/>	INTDATE	INTDATE(ANSI), INTDATE(LILIAN)	
<input type="checkbox"/>	LISTING	LIST, NOLIST	
<input type="checkbox"/>	MAP	MAP, NOMAP	
<input type="checkbox"/>	SOURCE	S, NOS	

Importing Build Tools

Purpose Follow this procedure to import a build tool from an external XML file.

To import a build tool:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the content area click Import Build Tool.



The Import Build Tools from External File dialog box appears.

- 3** Do one of the following:
 - Type the path to the file.
 - Click Browse, navigate to the file, select it, and click Open.
- 4** Click OK. The build tool is imported into Dimensions Build and added to the list in the content area.

Modifying Build Tools

Purpose Follow this procedure to modify the details of an existing build tool.

To modify a build tool:

- 1** In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2** In the navigation or content areas select the build tool that you want to modify. The content area refreshes and displays the details of the build tool.
- 3** On the Build Tools Details tab and Edit Build Tool Details.



The Edit Build Tool dialog box appears.

- 4** Modify the build tool name if required.
- 5** Modify the build tool description if required.
- 6** Click OK. The details of the build tool are updated in the content area.

Modifying Build Tool Options

Purpose Follow this procedure to modify the options for an existing build tool.

To modify build tool options:

- 1** In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2** In the navigation or content areas select the build tool containing the options that you want to modify. The content area refreshes and displays the details of the build tool.
- 3** In the Build Tool Possible Options section of the content area, click the build option that you want to modify. The Edit Build Tool Option dialog box appears.
- 4** Modify the option name if required.
- 5** Modify the option description if required.

- 6 Modify the option values and their descriptions if required. To add a new option, in the **Possible Values** field type a value for the option. In the **Descriptions of Values** field optionally type a description of the new value.
- 7 Click OK.

Deleting Build Tools

Purpose Follow this procedure to delete build tools. When you delete build tools their build options are also deleted.

To delete build tools:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the content areas select the check box next to the build tools that you want to delete.
- 3 Click Delete. To confirm that you want to delete the build options click Yes.

Deleting Build Tool Options

Purpose Follow this procedure to delete build options from a build tool. This procedure does not delete the parent build tool.

To delete build tool options:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools.
- 2 In the navigation or content areas select the build tool that contains the build options that you want to delete.
- 3 On the Build Tool Possible Options tab select the check boxes of all the build options that you want to delete.
- 4 Click Delete. To confirm that you want to delete the build tool options click Yes.

Build Option Groups

About Build Option Groups

Build option groups are logical collections of build tool options that are passed to the build engines and scripts that execute builds, and enables you to apply rules and enforce standards. Examples of option groups are `Compile_Options` and `Link_Options`.

About Build Option Properties

For each build option group you specify a preferred build tool and build options. Build options have the following properties:

- **Name:** the name of the option. Multiple build options can share the same name.
- **Value Type:** is one of the following:
 - Option Rule: enables you to select a build tool, and then specify options and values for that build tool.
 - Freeform Text: enables you to specify an option that is tailored to your requirements.
- (For Option Rule only) **Build Tool:** specifies the preferred build tool (does not have to be the preferred build tool specified for the build option group).
- (For Option Rule only) **Option Rule:** specifies an option rule for the build tool you select.
- **Value:**
 - If you selected Option Rule, specifies a value for the option rule.
 - If you selected Freeform text, specifies the value of the build option.

Build Option Group Examples

- **Common CICS COBOL Options**

Preferred build tool: IBM CICS Preprocessor V2R3

Build options:

Build Option	Value Type	Build Tool	Option Rule	Value
COBOL_OPTS	Option Rule	IBM COBOL Compiler v1.4	SOURCE	S
CICS_OPTS	Option Rule	IBM CICS Preprocessor V2R3	APOST	QUOTE

■ Common COBOL Options

Preferred build tool: IBM COBOL Compiler v1.4

Build options:

Build Option	Value Type	Build Tool	Option Rule	Value
COBOL_OPTS	Option Rule	IBM COBOL Compiler v1.4	SOURCE	S
COBOL_OPTS	Option Rule	IBM COBOL Compiler v1.4	INTDATE	INTDATE (LILLIAN)

About Build Options

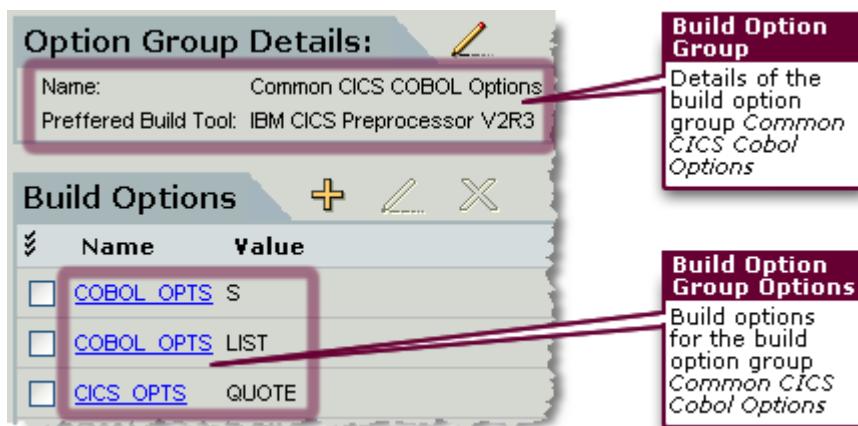
For details of the build options that you can add to build option groups, see *Dimensions Build Predefined Symbols* in *The Templating Language and Processor* chapter of the *Developer's Reference*.

Viewing Build Option Groups

Purpose Follow this procedure to view a list of build option groups and their options.

To view build option groups and options:

- 1 In the navigation area of the Build Management tab click Settings and click Build Option Groups. The content area lists all the build option groups that are currently defined.
- 2 To view a build option group's build options, in the **Name** column in the content area click the build option group name. The content area refreshes and lists all the build options defined for the build option group.



Adding Build Option Groups

Purpose Follow this procedure to add a new build option group.

To add a build option group:

- 1 In the navigation area of the Build Management tab click Settings and click Build Option Groups.
- 2 In the content or menu areas click New Object.



The Add New Build Option Group dialog box appears.

- 3 From the **Option Group Build Tool** list select a preferred build tool for the option group.
- 4 For **Option Group Name** type a name for the option group. Use meaningful names that reflect the context and usage of the option group, for example, *Common CICS COBOL Options* and *VC++ Compilers*.
- 5 Click OK. The new option group is added to the list of build option groups in the content area.

Adding Build Options to Build Option Groups

Purpose Follow this procedure to add a new build option to an existing build option group.

To add a build option to a build option group:

- 1 In the navigation area of the Build Management tab click Settings and click Build Option Groups.
- 2 In the navigation or content area click the build option group where you want to add a build option. The content area refreshes and displays the details of the build option group.
- 3 In the Build Options section of the content area click New Object.



The Add New Build Option Group Option dialog box appears.

- 4 For **Name** type a name for the option.
- 5 From the **Value Type** list select one of the following:
 - Option Rule: enables you to select a build tool, and then specify options and values.
 - Freeform Text: enables you to specify an option that is tailored to your specific requirements.

- 6 If you selected Option Rule from the Value Type list, do the following:
 - a From the **Build Tool** list select a build tool.
 - b From the **Option Rule** list select an option rule for the build tool. The list displays the possible option rules for the build tool that you selected from the Build Tool list.
 - c From the **Value** list select a value for the option rule. The list displays the possible values for the option rule that you selected from the Option Rule list. If the option rule does not have a default value, type your own.
- 7 If you selected Freeform Text from the Value Type list, in the **Value** field type the content of the build option.
- 8 Click OK. The new option is added to the list of build options for that build option group.

Modifying Build Option Groups

Purpose Follow this procedure to modify the details of an existing build option group.

To modify the details of build option groups:

- 1 In the navigation area of the Build Management tab click Settings and click Build Option Groups.
- 2 In the navigation or content areas select the build option group that you want to modify. The content area refreshes and displays the details of the option group.
- 3 In the Option Group Details section of the content area click Edit Option Group Details.



The Edit Option Group Details dialog box appears.

- 4 From the **Option Group Build Tool** list select a preferred build tool for the option group.
- 5 For **Option Group Name** modify the name of the option group. Use meaningful names that reflect the context and usage of the option group, for example, *Common CICS COBOL Options* and *VC++ Compilers*.
- 6 Click OK.

Modifying the Options of Build Option Groups

Purpose Follow this procedure to modify the build options of an existing build option group.

To modify the options of a build option group:

- 1** In the navigation area of the Build Management tab click Settings and click Build Option Groups.
- 2** In the navigation or content areas select the build option group containing the build options that you want to modify. The content area refreshes and displays the details of the build option group.
- 3** In the Build Options section of the content area click the build option that you want to modify. The Edit Build Option Group dialog box appears.
- 4** For **Name** modify the name of the option.
- 5** From the **Value Type** list select one of the following:
 - Option Rule: enables you to select a different build tool, and then specify options and values.
 - Freeform Text: enables you to specify an option that is tailored to your requirements.
- 6** If you selected Option Rule, do the following:
 - a** From the **Build Tool** list select a different build tool.
 - b** From the **Option Rule** list select an option rule for the build tool. The list displays the possible option rules for the build tool that you selected from the Build Tool list.
 - c** From the **Value** list select a value for the option rule. The list displays the possible values for the option rule that you selected from the Option Rule list. If the option rule does not have a default value, type your own.
- 7** If you selected Freeform Text, in the **Value** field modify the existing option or type the content of a new value.
- 8** Click OK.

Deleting Build Option Groups

Purpose Follow this procedure to delete build option groups and their associated build options.

To delete build option groups:

- 1** In the navigation area of the Build Management tab click Settings and click Build Option Groups.
- 2** In the content area select the check boxes next to the build option groups that you want to delete.
- 3** In the content or menu areas click Delete. To confirm that you want to delete the build option groups click Yes.

Deleting Options from Build Option Groups

Purpose Follow this procedure to delete build options from a build option group.

To delete options from a build option group:

- 1** In the navigation area of the Build Management tab click Settings and click Build Option Groups.
- 2** In the navigation or content areas click the build option group containing the build options that you want to delete.
- 3** In the Build Options section of the content area select the check boxes next to the builds option that you want to delete.
- 4** In the content or menu areas click Delete. To confirm that you want to delete the build options click Yes.

Transition Rule Templates

About Transition Rule Templates

A transition rule template describes a generic rule for building an item and applying it to specific source items at a later time. It describes how to change or transition an item of one file type (i.e., *.c) to an item of another file type (i.e., *obj) using a template or script.

A transition rule template is comprised of:

- The generic input mask(s) to the transition.
- The build tool used to perform the transition.
- The template or script used to execute the transition.
- The build options and option groups used during the transition (compile, link, etc.).
- The generic output mask of the transition.



NOTE Transition rule templates are not the same as build templates. For details about build templates and the templating language see the *Developer's Reference*.

Transition Rule Template Examples

■ ASM Compile

- Preferred build tool: IBM High Level Assembler
- Script/template: TEMPLATE(MDHBASC0)
- Input mask: ASM(*)
- Output mask: OBJ(*)
- Build options: DMMAXRC(4)

■ General Mainframe Program Link

- Preferred build tool: IBM Binder v1.4
- Script/template: TEMPLATE(LINKLE)
- Input masks: OBJ(*), SYSLIN(*)
- Output mask: LOAD(*)
- Build options: LINK_OPTS AMODE(31), LINK_OPTS CASE(MIXED)

About Build Options

For details of the build options that you can add to transition rule templates, see *Dimensions Build Predefined Symbols* in *The Templating Language and Processor* chapter of the *Developer's Reference*.

Viewing Transition Rule Templates

Purpose Follow this procedure to view a list of transition rule templates and their properties.

To view transition rule templates:

- 1 In the navigation area of the Build Management tab click Settings and click Build Tools. The navigation and content areas list all the transition rule templates that are currently defined.
- 2 To view the properties of a transition rule template, in the navigation or content areas click the transition rule template. The content area refreshes and displays the details of the transition rule template in the following tabs:
 - Inputs
 - Outputs
 - Build Options
 - Option Groups

To view the details of a specific property click the appropriate tab.

Adding Transition Rule Templates

Purpose Follow this procedure to add a new transition rule template.

To add a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the content or menu areas click New Object.



The Add New Transition Rule Template dialog box appears.

- 3 For **Name** type the name of the new transition rule template.
- 4 From the **Preferred Build Tool** list select the build tool that will be used to execute the transition. If you do not want to specify a build tool, select Undefined.

5 From the **Select the script type** list select one of the following:

- *Input script content manually*

In the **Script Content** field type the content of the script to be executed when this transition rule is run. To wrap the text select the **Word Wrap** check box.

- *Use Dimensions-controlled file as a script*

Choose a file from the Dimensions repository (see step 6 below).



NOTE If you are using a work area the file will be pushed to the build area. If you are using a deployment area, Dimensions Build assumes that the script has already been deployed to that area.

- *Use a file in the build area as a script*

Type the name of a script or template located in your build area.

- *Same as build configuration global script*

When you use a transition that has this script type the target is built using the main script defined for the parent build configuration. However, you can specify build options for the transition that are different to those defined in the main script. Therefore the build options that are used are those defined in the transition and not the main script.

For details about adding build options to transition rule templates see [page 176](#).

6 If you selected *Use Dimensions-controlled file as a script*, do the following:

- Click Browse. The Select File from Dimensions dialog box appears.
- In the left pane, expand Projects or Baselines and browse for the directory containing the file.
- In the right pane select the file.
- Click OK. The file is added to the Add New Transition Rule Template dialog box.

7 Click OK. The new transition rule template is added to the list of transition rule templates in the content area.

Modifying Transition Rule Templates

Purpose Follow this procedure to modify the details of an existing transition rule template.

To modify a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings and click Transition Rule Templates.
- 2 In the content area select the check box next to the transition rule template that you want to modify, and click Edit.



The Edit Transition Rule Template dialog box appears.

- 3 For **Name** modify the name of the transition rule template.
- 4 From the **Preferred Build Tool** list select the build tool used to execute the transition. If you do not want to specify a build tool, select Undefined.
- 5 From the **Select the script type** list select one of the following:
 - *Input script content manually*
In the **Script Content** field type or modify the content of the script to be executed when this transition rule is run. To wrap the text select the **Word Wrap** check box.
 - *Use Dimensions-controlled file as a script*
Choose a file from the Dimensions repository (see step 6 below).



NOTE If you are using a work area the file will be pushed to the build area. If you are using a deployment area, Dimensions Build assumes that the script has already been deployed to that area.

- *Use a file in the build area as a script*
Type or modify the name of a script or template in your build area.
- *Same as build configuration global script*
When you use a transition that has this script type the target is built using the main script defined for the parent build configuration. However, you can specify build options for the transition that are different to those defined in the main script. Therefore the build options that are used are those defined in the transition and not the main script.

For details about adding build options to transition rule templates see [page 176](#).

- 6 If you selected *Use Dimensions-controlled file as a script*, do the following:
 - a Click Browse. The Select File from Dimensions dialog box appears.
 - b In the left pane, expand Projects or Baselines and browse for the directory containing the file.
 - c In the right pane select the file.
 - a Click OK. The file is added to the Add New Transition Rule Template dialog box.
- 7 Click OK. The details of the transition rule template are updated in the content area.

Deleting Transition Rule Templates

Purpose Follow this procedure to delete transition rule templates.

To delete transition rule templates:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the content area select the check box next to each transition rule template that you want to delete.

- 3 In the content or menu areas click Delete.
- 4 To confirm that you want to delete the transition rule templates click Yes.

Adding Input Masks to Transition Rule Templates

Purpose Follow this procedure to add a new generic input mask (file type) to a transition rule template, for example, ASM(*).

To add an input mask to a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas select the transition rule template to which you want to add a new input mask. The content area refreshes and displays the details of the transition rule template.
- 3 In the content area click the Inputs tab, and click New Object.



The Add New Transition Rule Template Input Mask dialog box appears.

- 4 For **Mask Value** type a value for a new generic file type.
- 5 Click OK. The new mask value is added to the list on the Inputs tab.

Modifying Transition Rule Template Input Masks

Purpose Follow this procedure to modify the value of an existing generic input mask for a transition rule template.

To modify a transition rule template input mask:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas select the transition rule template containing the input mask that you want to modify. The content area refreshes and displays the details of the transition rule template.
- 3 In the content area click the Inputs tab, select the check box next to the input mask, and click Edit.



The Edit Transition Rule Template Input Mask dialog box appears.

- 4 For **Mask Value** modify the value of the generic file type.
- 5 Click OK. The modified mask value is updated in the list on the Inputs tab.

Deleting Input Masks from Transition Rule Templates

Purpose Follow this procedure to delete input masks from a transition rule template.

To delete input masks from a transition rule template:

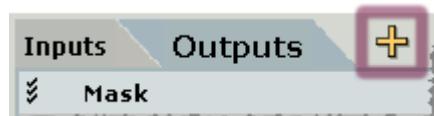
- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas click the transition rule template containing the input masks that you want to delete.
- 3 In the content area, click the Inputs tab.
- 4 Select the check box next to each input mask that you want to delete.
- 5 Click Delete.
- 6 To confirm that you want to delete the input masks click Yes.

Adding Output Masks to Transition Rule Templates

Purpose Follow this procedure to add a new generic output mask (file type) to a transition rule template, for example, OBJ(*) .

To add an output mask to a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas select the transition rule template to which you want to add a new input mask. The content area refreshes and displays the details of the transition rule template.
- 3 In the content area click the Outputs tab, and click New Object.



The Add New Transition Rule Template Output Mask dialog box appears.

- 4 For **Mask Value** type a value for a new generic file type.
- 5 Click OK. The new mask value is added to the list on the Outputs tab.

Modifying Transition Rule Template Output Masks

Purpose Follow this procedure to modify the value of an existing generic output mask for a transition rule template.

To modify a transition rule template output mask:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas select the transition rule template containing the output mask that you want to modify. The content area refreshes and displays the details of the transition rule template.
- 3 In the content area click the Outputs tab, select the check box next to the output mask, and click Edit.



The Edit Transition Rule Template Output Mask dialog box appears.

- 4 For **Mask Value** modify the value of the generic file type.
- 5 Click OK. The modified mask value is updated in the list on the Outputs tab.

Deleting Output Masks from Transition Rule Templates

Purpose Follow this procedure to delete output masks from a transition rule template.

To delete output masks from a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas click the transition rule template containing the output masks that you want to delete.
- 3 In the content area, click the Outputs tab.
- 4 Select the check box next to each output mask that you want to delete.
- 5 Click Delete.
- 6 To confirm that you want to delete the output masks click Yes.

Adding Build Options to Transition Rule Templates

Purpose Follow this procedure to add a new build option to a transition rule template.

To add a build option to a transition rule template:

- 1** In the navigation area of the Build Management tab click Settings and click Transition Rule Template.
- 2** In the navigation or content area click the transition rule template where you want to add a build option. The content area refreshes and displays the details of the transition rule template.
- 3** In the Build Options section of the content area click New Object.



The Add New Transition Rule Template Option dialog box appears.

- 4** For **Name** type a name for the option.
- 5** From the **Value Type** list select one of the following:
 - *Freeform Text*
In the **Value** field type an option that is tailored to your specific requirements.
 - *Option Rule*
Select a build tool and then select an option rule and a value (see step 6).
 - *Option Link*
Select a build option group and then select an option (see step 7).
- 6** If you selected Option Rule from the Value Type list, do the following:
 - a** From the **Build Tool** list select a build tool. If you do not want to specify a build tool select *Undefined*.
 - b** (Not applicable if you selected *Undefined*) From the **Option Rule** list select an option rule for the build tool. The list displays the possible option rules for the build tool that you selected from the Build Tool list.
 - c** (Not applicable if you selected *Undefined*) From the **Value** list select a value for the option rule. The list displays the possible values for the option rule that you selected from the Option Rule list. If the option rule does not have a default value, type your own.
- 7** If you selected Option Link from the Value Type list, do the following:
 - a** From the **Group** list select a build option group.
 - b** From the **Option** list select an option for the build option group. The list displays the possible options for the build option group that you selected from the Group list.
- 8** Click OK. The new build option is added to the list of build options for the transition rule template.

Modifying Build Options for Transition Rule Templates

Purpose Follow this procedure to modify a build option for a transition rule template.

To modify a build option for a transition rule template:

- 1** In the navigation area of the Build Management tab click Settings and click Transition Rule Template.
- 2** In the navigation or content area click the transition rule template containing the build option that you want to modify. The content area refreshes and displays the details of the transition rule template.
- 3** In the Build Options section of the content area click the build option that you want to modify. The Edit Transition Rule Template Option dialog box appears.
- 4** For **Name** modify the name of the option.
- 5** From the **Value Type** list select one of the following:
 - *Freeform Text*
In the **Value** field type an option that is tailored to your specific requirements, or modify the existing option.
 - *Option Rule*
Select a different build tool and then select an option rule and a value (see step 6).
 - *Option Link*
Select a different build option group and then select an option (see step 7).
- 6** If you selected Option Rule from the Value Type list, do the following:
 - a** From the **Build Tool** list select a different build tool. If you do not want to specify a build tool select *Undefined*.
 - b** (Not applicable if you selected *Undefined*) From the **Option Rule** list select an option rule for the build tool. The list displays the possible option rules for the build tool that you selected from the Build Tool list.
 - c** (Not applicable if you selected *Undefined*) From the **Value** list select a value for the option rule. The list displays the possible values for the option rule that you selected from the Option Rule list. If the option rule does not have a default value, type your own.
- 7** If you selected Option Link from the Value Type list, do the following:
 - a** From the **Group** list select a different build option group.
 - b** From the **Option** list select an option for the build option group. The list displays the possible options for the build option group that you selected from the Group list.
- 8** Click OK. The modified build option is updated in the list of build options for the transition rule template.

Deleting Build Options from Transition Rule Templates

Purpose Follow this procedure to delete build options from a transition rule template.

To delete build options from a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas click the transition rule template containing the build options that you want to delete.
- 3 In the content area, click the Build Options tab.
- 4 Select the check box next to each build option that you want to delete.
- 5 Click Delete.
- 6 To confirm that you want to delete the build options click Yes.

Adding Build Option Groups to Transition Rule Templates

Purpose Follow this procedure to add a build option group to a transition rule template. You can only add build option groups that are already defined. For details about adding new build option groups see [page 165](#).

To add a build option group to a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings and click Transition Rule Template.
- 2 In the navigation or content area click the transition rule template where you want to add a build option group. The content area refreshes and displays the details of the transition rule template.
- 3 In the content area click the Option Groups tab, and click New Object.



The Select Transition Rule Template Build Option Group dialog box appears.

- 4 Select a build option group from the list.
- 5 Click OK. The build option group is added to the list of build option groups for the transition rule template.

Deleting Build Option Groups from Transition Rule Templates

Purpose Follow this procedure to delete build option groups from a transition rule template.

To delete build option groups from a transition rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Transition Rule Templates.
- 2 In the navigation or content areas click the transition rule template containing the build options that you want to delete.
- 3 In the content area, click the Option Groups tab.
- 4 Select the check box next to each build option group that you want to delete.
- 5 Click Delete.
- 6 To confirm that you want to delete the build option groups click Yes.

Application Rule Templates

About Application Rule Templates

An application rule template defines a group of transition rule templates that are used to create an application.



NOTE Application rule templates are not the same as build templates. For details about build templates and the templating language see the *Developer's Reference*.

Application Rule Template Examples

- **CICS COBOL Program w/Assembler**

Input Mask	Transition Rule Template	Output Mask
COBOL()	COBOL Compile	OBJ()
COBOL()	CICS COBOL Compile	OBJ()
ASM()	Assembler Compile	OBJ()
OBJ()	General Program Link	LOAD()

- **MS .NET Executable**

Input Mask	Transition Rule Template	Output Mask
stdafx.cpp	MS C++ Compilation Creating PCH	stdafx.obj, stdafx.pch
*.cpp, *.pch	MS C++ Compilation Using PCH	*.obj
*.obj	MS C/C++ Linkage	*.exe

Viewing Application Rule Templates

Purpose Follow this procedure to view a list of all the application rule templates currently defined.

To view application rule template:

- 1 In the navigation area of the Build Management tab click Settings and click Application Rule Templates. The content area lists all the application rule templates that are currently defined.
- 2 To view the properties of an application rule template, in the navigation or content areas click the application rule template. The content area refreshes and lists all of the current properties.

Adding Application Rule Templates

Purpose Follow this procedure to add a new application rule template.

To add an application rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Application Rule Templates.
- 2 In the content or menu areas click New Object.



The Add New Application Rule Template dialog box appears.

- 3 For **Name** type a name for the new application rule template
- 4 For **Description** type a description for the new application rule template.
- 5 Click OK. The new application rule template is added to the list of application rule templates in the content area.

Modifying Application Rule Templates

Purpose Follow this procedure to modify an application rule template.

To modify an application rule template:

- 1 In the navigation area of the Build Management tab click Settings, and click Application Rule Templates.
- 2 In the content area select the check box next to the application rule template that you want to modify, and click Edit.



The Edit Application Rule Template dialog box appears.

- 3** For **Name** modify the name of the application rule template.
- 4** For **Description** modify the description of the application rule template.
- 5** Click OK. The details of the application rule template are updated in the content area.

Deleting Application Rule Templates

Purpose Follow this procedure to delete application rule templates.

To delete application rule templates:

- 1** In the navigation area of the Build Management tab click Settings, and click Application Rule Templates.
- 2** In the content area select the check box next to each application rule template that you want to delete, and click Delete.
- 3** To confirm that you want to delete the application rule templates click Yes.

Adding Transition Rule Templates to Application Rule Templates

Purpose Follow this procedure to add transition rule templates to an application rule template.



NOTE If you update a transition rule template its details are also updated in any application rule template where it is included.

To add transition rule templates to an application rule template:

- 1** In the navigation area of the Build Management tab click Settings, and click Application Rule Template.
- 2** In the content or menu areas select the application rule template where you want to add a transition rule template. Click New Object.



The Assign Target Transition Rule Templates dialog box appears.

- 3** From the **Select Transition Rule Templates** list select one or more transition rule templates.
- 4** Click OK. The transition rule templates are added to the content area.

Removing Transition Rule Templates from Application Rule Templates

Purpose Follow this procedure to remove transition rule templates from application rule templates. This action does not delete the transition rule templates from Dimensions Build.

To remove transition rule templates from an application rule template:

- 1** In the navigation area of the Build Management tab click Settings, and click Application Rule Templates.
- 2** In the navigation or content areas select the application rule template containing the transition rule templates that you want to remove.
- 3** In the content area, select the check box next to each transition rule template that you want to remove.
- 4** Click Delete.
- 5** To confirm that you want to remove the transition rule templates click Yes.

Chapter 8

Managing Build Configurations

About Build Configurations	184
About Build Projects	184
Basic Build Configuration Operations	186
Build Areas	191
Sources	198
Build Targets	202
Inputs	206
Outputs	209
Scripts	210
Build Options	219
Versions	222
Exporting and Importing	222
Tips	225

About Build Configurations

A build configuration captures information about what will be built, and how it will be built. Build configurations include the following:

- **Build Areas**

A build area defines where the files related to a build will be checked out to, or delivered to. Although build areas are created in the Dimensions Administration Console (and referred to there as file areas), each build configuration must specify one or more build areas. Most builds will use work areas or deployment areas.

- **Build Targets**

A build target defines the end-product of a build. Build targets include the specifications for inputs and outputs. Build targets can be real, such as an executable file, or virtual, such as "clean" or "all".

- **Sources**

A source can be a source code file, an include file, another target, and so on. A source file must be imported as a source and then defined as an input to a target.

- **Build Options**

A build option defines elective settings that may or may not be part of the build, such as a debug switch. Individual options may be collected into option groups. For details of the build options see *Dimensions Build Predefined Symbols* in *The Templating Language and Processor* chapter of the *Developer's Reference*.

- **Scripts**

A script defines the specific commands necessary to execute the build. Scripts indicate which build tool is to be used, which build options, and so on. You can define scripts to execute before the build, during the build, and after the build.

- **Versions**

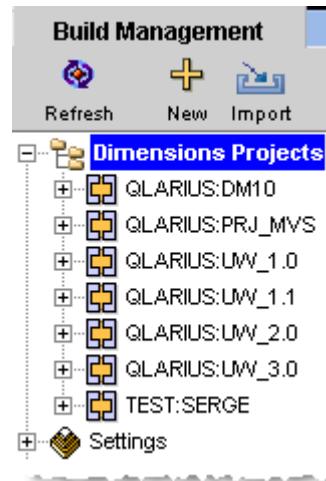
A version is a previously-checked-in build configuration. Versions are numbered, time- and date-stamped, and include associations with build areas and targets.

Build configurations can be exported to an XML file (and later re-imported).

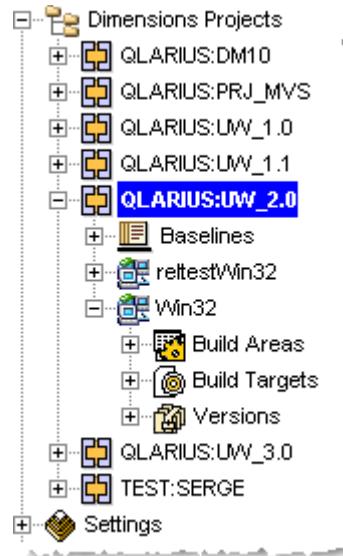
About Build Projects

Dimensions organizes build configurations underneath Dimensions projects; that is, projects defined in Dimensions itself. Part of the process of creating a build configuration is to select a Dimensions project.

The navigation area of the Build Management tab shows the Dimensions projects in use.



If no projects have been associated with a build configuration yet, no Dimensions projects will appear in the tree. Expanding a specific project reveals the build configurations, and their components, underneath:



Similar, but not identical, project and configuration trees appear under the Scheduling, Monitoring, and Notifications tabs. Since those functions are primarily described in sections dedicated to those tasks, this section will describe the behavior underneath the Build Management tab.

Basic Build Configuration Operations

Viewing Build Configurations

To view the existing build configurations, use the following procedure.

To view build configurations:

- 1** In the Dimensions Administration Console, click the Build Management tab.
- 2** Expand the Dimensions Projects tree.
- 3** Select a project. The Build Configurations tab appears, displaying the available build configurations for that project.

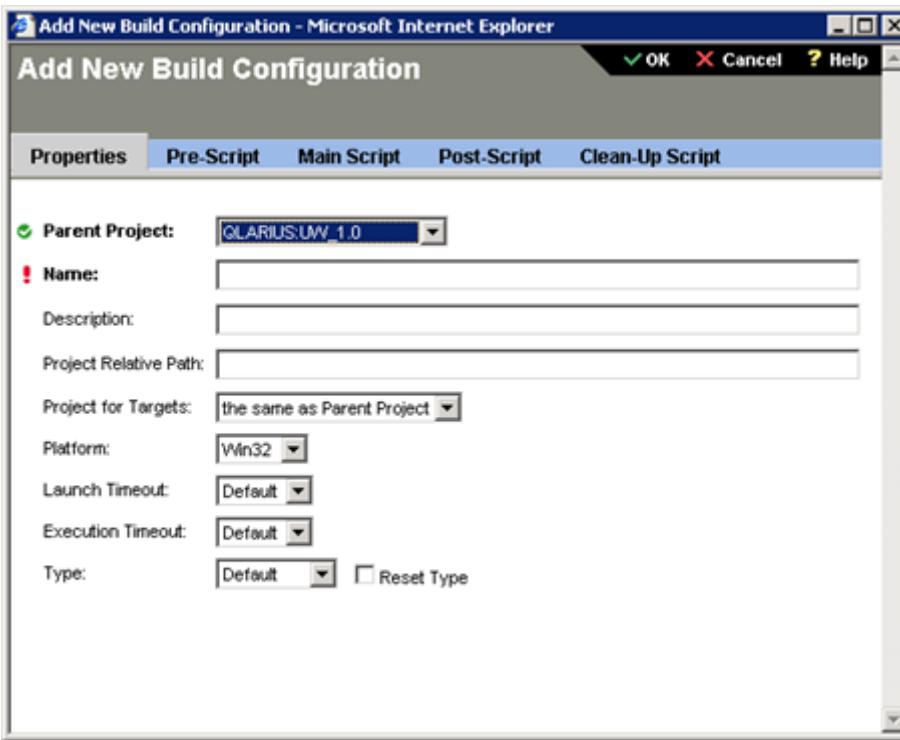
Creating Build Configurations

To create a new build configuration, use the following procedure.

To create a new build configuration:

- 1** If your build configuration requires the creation of a new work area or deployment area, create the area first as described in [Build Areas](#) on page 191.
- 2** In the Dimensions Administration Console, click the Build Management tab.

- 3** Click the New Object icon (shaped like a plus sign). The Add New Build Configuration dialog box appears.



NOTE After you create at least one build configuration, associated Dimensions projects appear in the navigation tree. Once Dimensions projects appear, you can select one before clicking the New Object icon. The selected project will appear in the Parent Project field.

- 4** Use the Properties tab to set the project details and timeout details, as described in [Setting Build Configuration Properties on page 190](#).
- 5** If you wish to create or select a script to be executed before the build is launched (but after source files have been transferred to the build area), use the Pre-Script tab. See [Adding a Script on page 213](#).
- 6** If you have multiple targets that can use the same script, create or select a main script and then set each target to **Use build configuration main script**. Note that any transition scripts for individual targets will take precedence over the main script. See [Adding a Script on page 213](#).
- 7** If you wish to create or select a script to be executed after the build is launched, use the Post-Script tab. See [Adding a Script on page 213](#).
- 8** If you wish to create or select a script to be executed before the build is launched (and before source files are transferred to the build area), use the Clean-Up Script tab. This script is executed if you select **Execute clean-up script before build** when running a build. See [Adding a Script on page 213](#).

Note that creating a build configuration does not require you to add the build target or source files for the project. That can be done later.

Modifying Build Configurations

To modify an existing build configuration, there are two procedures, depending on whether you want to modify the properties and scripts, or the build areas, targets, sources, and options.

To modify the properties or scripts of an existing build configuration:

- 1 In the Dimensions Administration Console, select the Build Management tab if it is not selected already.
- 2 Expand the Dimensions Projects tree until you can see the build configuration that you want to modify.
- 3 Select the build configuration. The Build Configuration Details, Build Areas, and other related sections appear in the content area.
- 4 Check out the build configuration if it is not checked out already.
- 5 Click the Edit/View icon (shaped like a pencil).



The Edit Build Configuration dialog box appears.

- 6 Use the Properties tab to edit the project details and timeout details, as described in [Setting Build Configuration Properties](#) on page 190.
- 7 Use the Pre-Script tab to edit or specify the details of the script, if any, that is to be executed before the build is launched (but after source files have been transferred to the build area). See [Modifying a Build Configuration Script](#) on page 216.
- 8 Use the Main Script tab to edit or specify the details of the main script. See [Modifying a Build Configuration Script](#) on page 216.
- 9 Use the Post-Script tab to edit or specify the details of the script, if any, that is to be executed after the build is launched. See [Modifying a Build Configuration Script](#) on page 216.
- 10 Use the Clean-Up Script tab to edit or specify the details of the script, if any, that is to be executed before the build is launched (and before any source files are transferred into the build area). See [Modifying a Build Configuration Script](#) on page 216.

To modify the build configuration details, build areas, targets, sources, build options, or build option groups associated with a build configuration:

- 1 In the Dimensions Administration Console, click the Build Management tab.
- 2 Expand the Dimensions Projects tree until you can see the build configuration that you want to modify.
- 3 Check out the build configuration if it is not checked out already.
- 4 Select the project containing the build configuration. The Build Configurations Details section appears, along with the tabs for Build Areas, Build Targets, and Build Options.

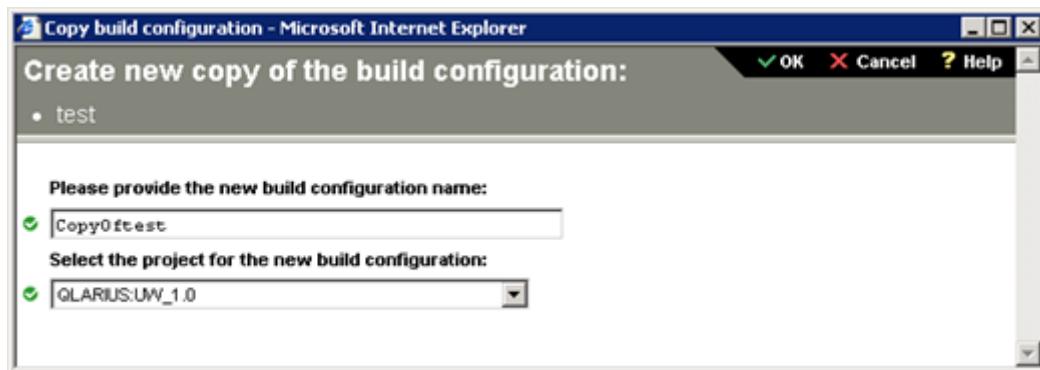
- 5** To edit the project details and timeout details, click the pencil icon next to the Build Configuration Details section title. Make the desired changes as described in [Setting Build Configuration Properties](#) on page 190.
- 6** To change a build area used by the build configuration, select the build area and click the pencil icon next to the Build Areas section title. Make the desired changes as described in [Modifying Build Areas](#) on page 197.
- 7** To change a build target used by the build configuration, select the build target and click the pencil icon next to the Build Targets section title. Make the desired changes as described in [Modifying Build Targets](#) on page 205.
- 8** To change a build source used by the build configuration, click the Sources tab, then select the check box next to the build source to be edited and click the pencil icon next to the Build Targets section title. Make the desired changes as described in [Modifying Build Sources](#) on page 201.
- 9** To change a build option used by the build configuration, select the build option and click the pencil icon next to the Build Options section title. Make the desired changes as described in [Modifying Build Options](#) on page 221.
- 10** To change a build option group used by the build configuration, click the Option Groups tab, then select the check box next to the build option group to be edited and click the pencil icon next to the Option Groups section title. Make the desired changes as described in [Modifying Build Option Groups](#) on page 166.

Copying Build Configurations

Copying an existing build configuration and then modifying the copy is an efficient way of creating a new build configuration, or even multiple new build configurations.

To copy a build configuration:

- 1** In the Dimensions Administration Console, click the Build Management tab.
- 2** Expand the Dimensions Projects tree until you can see the build configuration that you want to copy.
- 3** Select the build configuration that you want to copy.
- 4** Click the Copy icon. The **Create new copy of the build configuration** dialog box appears.



- 5** Enter the name of the new build configuration.

- 6 Select the project to receive the new build configuration.
- 7 Click OK. A confirmation dialog briefly displays, and the new build configuration appears in the Build Configurations section.

Deleting Build Configurations

If you want to delete a build configuration, you can do so using the following procedure.

To delete a build configuration:

- 1 In the Dimensions Administration Console, click the Build Management tab.
- 2 Expand the Dimensions Projects tree until you can see the build configuration that you want to delete.
- 3 Select the project containing the build configuration. The Build Configurations section appears.
- 4 Select the check box next to the build configuration that you want to delete. If there is more than one in the same project, you can select them all.
- 5 Click the Delete icon (shaped like an X). The Delete Build Configurations dialog box appears.
- 6 Click Yes to confirm the deletion.

Setting Build Configuration Properties

The Properties tab makes it possible for you to edit the details of a build configuration.

To create or edit the information on the Properties tab:

- 1 Edit the desired properties from the list below:
 - Parent Project
Select the Dimensions project to associate with the build configuration.
 - Name
Enter a name for the build configuration.
 - Description
Enter a description for the build configuration.
 - Project Relative Path
Enter a path that is relative to the Dimensions project root directory. This restricts the build to the sub-directory that you specify. For example, if the project root directory is C:\Projects\Qlarius and you specified a relative path of Java the build will be restricted to C:\Projects\Qlarius\Java.
 - Project for Targets
Select a Dimensions project to receive the target, or accept the default value of **the same as Parent Project**.

- Platform
Specify the platform (Win32, Linux, and so on) for the build configuration.
- Launch Timeout
Select Default, None, or Custom. This value represents an interval in seconds before Dimensions will abandon a build job that is stalled in its launch attempt. If you select **Custom**, an additional field appears, letting you enter a value in seconds.



NOTE On MVS, the usual value is None and you should be careful when selecting any other value. An incorrect configuration could result in a days-long delay.

- Execution Timeout
Select Default, None, or Custom. This value represents an interval in seconds before Dimensions will abandon a build job that is stalled during the execution of the build script. If you select **Custom**, an additional field appears, letting you enter a value in seconds.
- Type
Select Default or Openmake.
- Reset Type
Useful only if **Type** is changed from Default to Openmake. If selected, this check box causes settings required for an Openmake build configuration to be restored.

- 2 Click OK to accept the dialog box.
- 3 Click the Check In icon to check in the build configuration.

Build Areas

About Build Areas

Every build configuration in Dimensions must be associated with at least one build area. The three types of build areas are:

- Work areas
Use this type of area to work on files checked out from Dimensions. Defining a work area indicates that you want to have files automatically placed there.
- Deployment areas
Use this type of area to receive files that have been promoted to a chosen state in the Global Stage Lifecycle. For example, when a set of files is ready for System Test, they can be copied to an area that Quality Assurance uses.



NOTE Deployment scripts should be placed in the directory \$DM_ROOT/templates on the node that hosts the area. Do not place the deployment scripts in the area itself.

- Library cache areas

Use this type of area to receive the latest (tip) revision of a project. This is useful for providing copies of a project's files to other groups. In general, this type of area is not used to create builds.



NOTE Build areas are created in the Dimensions Administration Console, although they are referred to there as file areas. See the *Serena Dimensions CM Process Modeling User's Guide* for more information about file areas.

Creating a New Work Area

To create a new work area:

- 1 Log in to the Dimensions Administration Console.
- 2 Under the Distributed Development heading, click **Area Definitions**. The Area Definitions page appears.
- 3 Click the New icon. The pop-up menu displays the types of area that you can create:
 - Work area
 - Deployment area
 - Library cache area
- 4 Select **Work Area**. The New Work Area dialog box appears:

Available Users / Groups:		>>	Users / Groups granted permission to access this area:	
DMSYS	USER2	USER3	USER4	USER5
USER6	ADMIN			

- 5** Fill out the fields as follows:

- **ID**

The identifier for the build area.

- **Description**

Enter a description for the build area.

- **Network Node**

The name of the machine that Dimensions is to use when executing the build.

- **Directory**

The name of the directory that Dimensions is to use as the base directory; relative pathnames will be evaluated based on this directory.

- **Node User ID**

The user ID that Dimensions is to use when executing the build.



NOTE On Windows 200x platforms, the user should have administrator privileges. On Windows XP, or on UNIX, USS, or MVS, administrator privileges are not required.

- **Node Password**

The user password that Dimensions is to use when executing the build.

- **Area Owner**

This is the user ID that has permission to edit the build area.

- **Available Users/Groups and Users/Groups granted permission to access this area**

Select entries from the Available Users/Groups and use the >> and << links to move the selected entries to the list of users/groups with permission to access the build area.

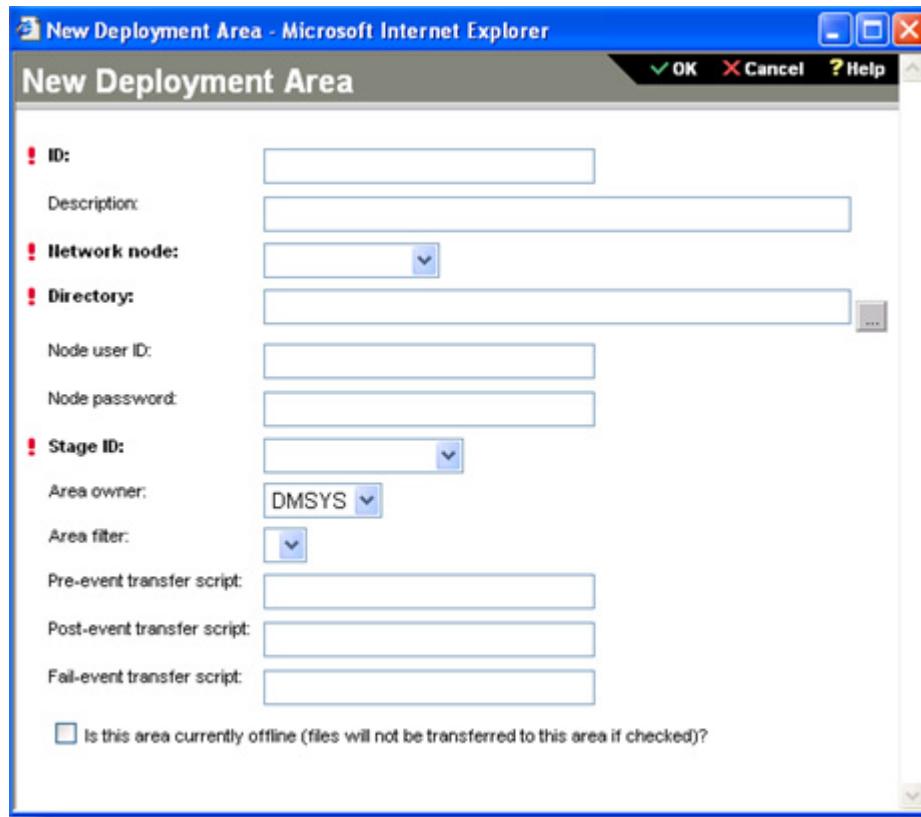
- 6** When you have finished filling out the dialog box, click **OK**. A Results dialog box briefly confirms the creation of the new area. The new area also appears in the File Areas List.

Creating a New Deployment Area

To create a new deployment area:

- 1** Log in to the Dimensions Administration Console.
- 2** In the upper right-hand corner, click on Area Definitions. The Area Definitions page appears.
- 3** Click the New icon. The pop-up menu displays the types of area that you can create:
 - Work area
 - Deployment area
 - Library cache area

- 4 Select Deployment area. The New Deployment Area dialog box appears:



- 5 Fill out the fields as follows:

- ID
The identifier for the build area.
- Description
Enter a description for the build area.
- Network Node
The name of the machine that Dimensions is to use when executing the build.
- Directory
The name of the directory that Dimensions is to use as the base directory; relative pathnames will be evaluated based on this directory.
- Node User ID
The user ID that Dimensions is to use when executing the build.



NOTE On Windows 200x platforms, the user should have administrator privileges. On Windows XP, or on UNIX, USS, or MVS, administrator privileges are not required.

- Node Password
The user password that Dimensions is to use when executing the build. This user should have Administrator privileges.

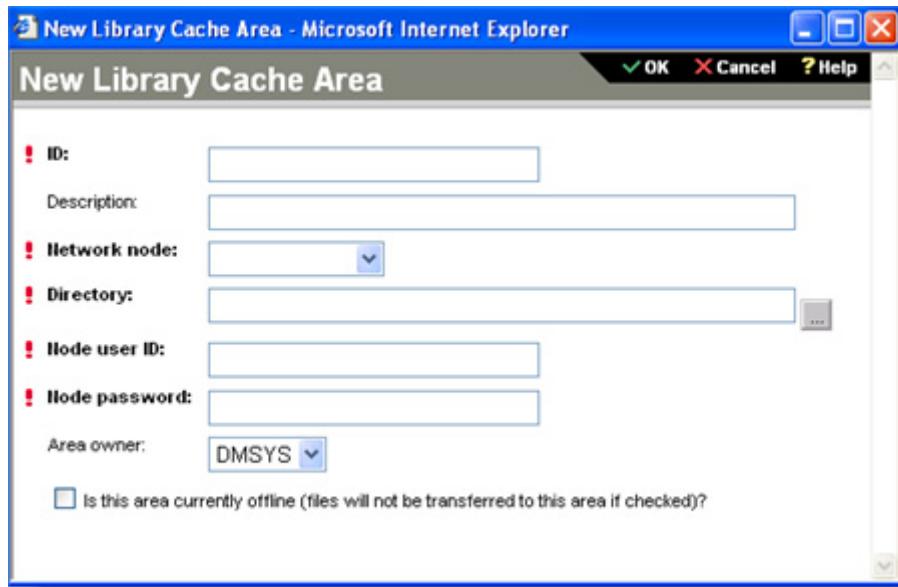
- Stage ID
When items reach the lifecycle stage specified here, This is the lifecycle stage, named in Dimensions, that will trigger the movement of files.
 - Area Owner
This is the user ID that has permission to edit the build area.
 - Filter
This is a template that determines which files will be taken up by the build. For example, a filter defined as "*.java" would only accept Java source files.
 - Pre-event Transfer Script
This is the name of a script to be executed before the files are transferred to the deployment area.
 - Post-event Transfer Script
This is the name of a script to be executed after the files are transferred to the deployment area.
 - Fail-event Transfer Script
This is the name of a script to be executed if the operation to transfer the files to the deployment area fails.
 - Is this area currently offline
If selected, this check box will cause the area to be excluded from file transfer operations. (You might want to do this to avoid overwriting a specific directory.)
- 6** When you have finished filling out the dialog box, click OK. A Results dialog box briefly confirms the creation of the new area. The new area also appears in the File Areas List.

Creating a New Library Cache Area

To create a new library cache area:

- 1** Log in to the Dimensions Administration Console.
- 2** In the upper right-hand corner, click Area Definitions. The Area Definitions page appears.
- 3** Click the New icon. The pop-up menu displays the types of area that you can create:
 - Work area
 - Deployment area
 - Library cache area

- 4 Select Library cache area. The New Library Cache Area dialog box appears:



- 5 Fill out the fields as follows:

- ID
The identifier for the build area.
- Description
Enter a description for the build area.
- Network Node
The name of the machine that Dimensions is to use when executing the build.
- Directory
The name of the directory that Dimensions is to use as the base directory; relative pathnames will be evaluated based on this directory.
- Node User ID
The user ID that Dimensions is to use when executing the build.



NOTE On Windows 200x platforms, the user should have administrator privileges. On Windows XP, or on UNIX, USS, or MVS, administrator privileges are not required.

- Node Password
The user password that Dimensions is to use when executing the build.
- Area Owner
This is the user ID that has permission to edit the build area.
- Is this area currently offline
If selected, this check box will cause the area to be excluded from file transfer operations. (You might want to do this to avoid overwriting a specific directory.)

- 6 When you have finished filling out the dialog box, click OK. A Results dialog box briefly confirms the creation of the new area. The new area also appears in the File Areas List.

Modifying Build Areas

You can modify a build area after it has been created. However, once a build area is associated with a build configuration, you cannot modify the following properties:

- Area ID
- Stage ID
- Network node
- Directory

To modify a build area:

- 1 From the Dimensions Administration Console, click Area Definitions.
- 2 Click the name of the build area you want to modify. The Edit Build Area dialog box appears.
- 3 Modify the details that you want to change.
- 4 Click OK to accept the dialog box.

Deleting Build Areas

If you have the privileges to create a build area, you can also delete the build area.



NOTE Once the build area has been included in a checked-in build configuration, that build area can no longer be deleted. If the build configuration has not been checked in, you can detach the build area and then delete. If the build configuration has been checked in, you can also delete the entire build configuration.

To delete a build area:

- 1 From the Dimensions Administration Console, click Area Definitions.
- 2 Place a check mark next to the build area you want to delete.
- 3 Click the Delete icon. The Delete Area dialog box appears.
- 4 Click OK to confirm the deletion. Dimensions displays a brief confirmation dialog box, and refreshes the list of build areas.

Sources

Sources are files or other items needed to produce a target. To define a target such as an executable file, you specify which source files, include files, and other items are needed to create the target.

You can define sources at any time after a build configuration has been created.

You can add build sources from a file on disk, or from a file already archived in Dimensions.

Adding a Build Source from a File

To add a build source from a file:

- 1 In the navigation area of the Build Management tab, locate and select the build configuration to which you want to add a source.
- 2 Make sure the build configuration is checked out.
- 3 Expand the build configuration until the Build Targets entry is visible in the project tree. This will also cause the Sources section to appear in the content area.
- 4 In the content area, click the Sources title to make it active.
- 5 Click the New Object icon (shaped like a plus sign). The Add New Source dialog box appears.
- 6 Enter the name and location for the source, then click OK. The new source appears in the Sources section.

Using Wildcard Patterns to Add Build Sources from Dimensions



NOTE This feature is not available on MVS.

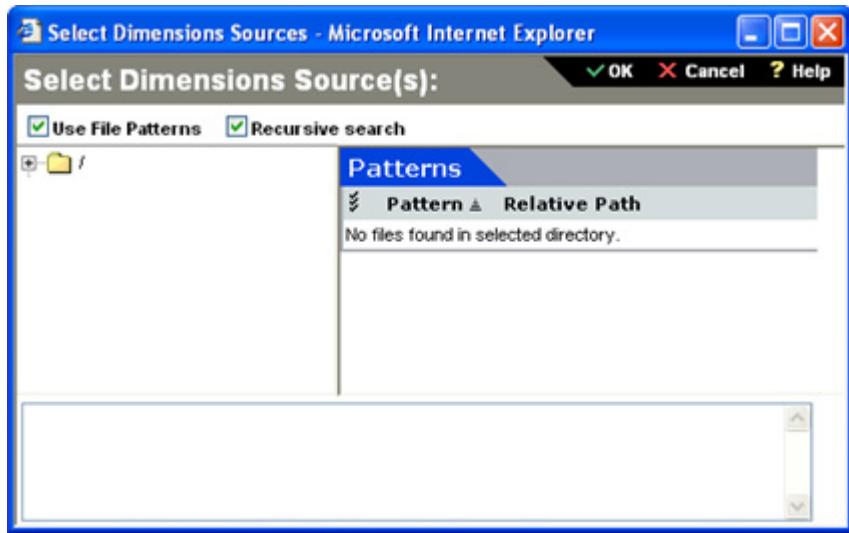
When you add a build source from Dimensions, you can add either individual files or wildcard patterns representing multiple files. An example of a wildcard pattern is *.java.

When you have to add many files, it is much easier to use wildcard patterns. Use the following procedure.

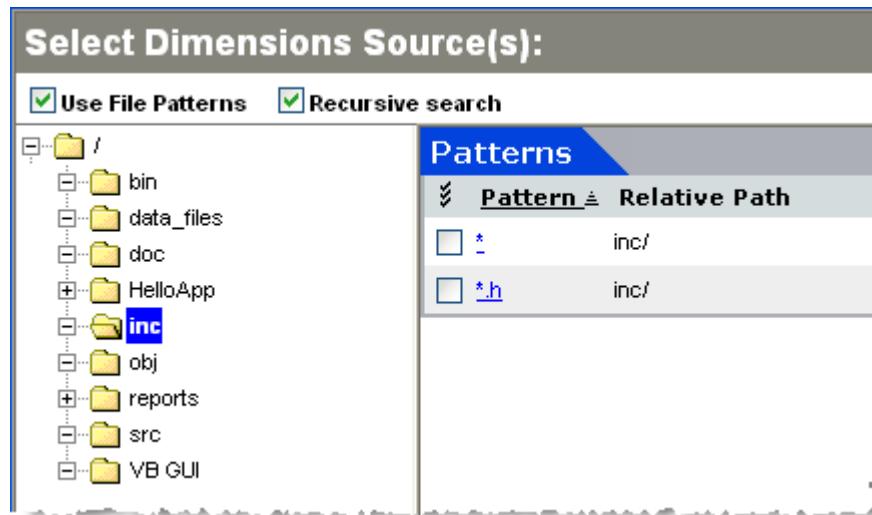
To add build sources from Dimensions, using wildcard patterns:

- 1 In the navigation area of the Build Management tab, locate and select the build configuration to which you want to add a source.
- 2 Make sure the build configuration is checked out.
- 3 Expand the build configuration until the Build Targets entry is visible in the project tree. This will also cause the Sources section to appear in the content area.
- 4 In the content area, click the Sources title to make it active.

- 5 Click the Add Sources from Dimensions icon. The Select Dimensions Sources dialog box appears.



- 6 Expand the directory tree. The folders displayed correspond to the folder structure for the associated Dimensions project. As you select each folder, a wildcard pattern appears in the Patterns area:



- 7 Select the desired wildcard pattern. Notice that the pattern appears at the bottom of the dialog box.



NOTE To select the entire contents of a directory, select the pattern "*". This pattern appears as long as there is at least one file in the directory.

- 8 Repeat with other folders until you have selected all desired wildcard patterns.
- 9 If you need to select individual files as well, use the procedure described in [Adding Individual Files in Dimensions as Build Sources on page 200](#). It is fine to switch back and forth between the two methods.

- 10** Click OK to accept the dialog box. The selected items appear in the Sources area of the Build Management tab.



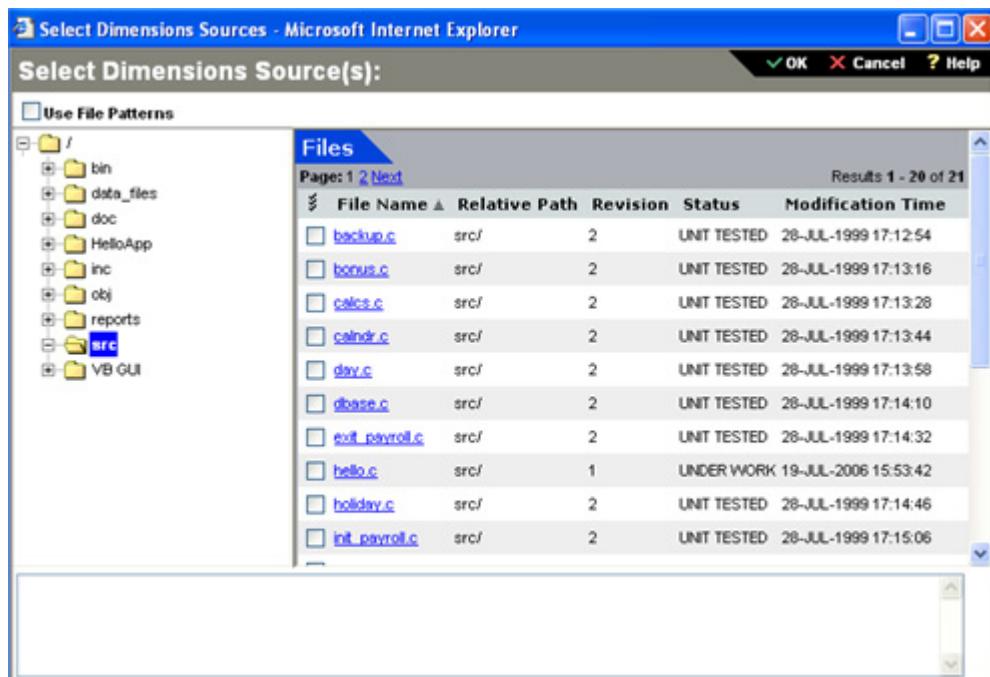
NOTE By default, Recursive Search is selected whenever Use File Patterns is selected. This means that a search will search throughout a directory tree. See the section on [Recursive Search](#), on page [201](#), for more information.

Adding Individual Files in Dimensions as Build Sources

In some situations you may want to add individual files as sources. To do this, you should deselect the Use File Patterns checkbox in the Select Dimensions Sources dialog box.

To add individual files in Dimensions as build sources:

- 1** In the navigation area of the Build Management tab, locate and select the build configuration to which you want to add a source.
- 2** Make sure the build configuration is checked out.
- 3** Expand the build configuration until the Build Targets entry is visible in the project tree. This will also cause the Sources section to appear in the content area.
- 4** In the content area, click the Sources title to make it active.
- 5** Click the Add Sources from Dimensions icon. The Select Dimensions Sources dialog box appears.
- 6** Deselect the Use File Patterns checkbox. The Patterns area disappears and a Files area replaces it. When you select a folder, the individual file listings appear:



- 7 Select the desired files. Notice that the file specifications appear at the bottom of the dialog box.
- 8 If you need to select files using a wildcard pattern also, use the procedure described in [Using Wildcard Patterns to Add Build Sources from Dimensions on page 198](#). It is fine to switch back and forth between the two methods.
- 9 Click OK to accept the dialog box. The selected items appear in the Sources area of the Build Management tab.

Recursive Search



NOTE Recursive search is not available on MVS.

In the Select Dimensions Sources dialog box, there is a checkbox for Recursive Search. By default, it is selected whenever Use File Patterns is selected:

When Recursive Search is on, Dimensions will generate wildcard patterns that will match an entire directory tree, or a file anywhere in the directory tree. For example:

`**/*@src/`

If Recursive Search is turned off, Dimensions will generate wildcard patterns that will search only in the current directory:

`*@src/`

Recursive Search is also disabled if you deselect Use File Patterns.

Modifying Build Sources

To modify an existing build source:

- 1 In the navigation area of the Build Management tab, locate and select the build configuration containing the source you want to modify.
- 2 Make sure the build configuration is checked out.
- 3 Expand the build configuration until the Build Targets entry is visible in the project tree. This will also cause the Sources section to appear in the content area.
- 4 In the content area, click the Sources title to make it active.
- 5 Click the Edit Selected Object icon (shaped like a pencil). The Edit Source dialog box appears.
- 6 Revise the source information, then click OK. The edited source appears in the Sources section.

Deleting Build Sources

To delete an existing build source or sources:

- 1 Locate the build configuration containing the source you want to delete.
- 2 Make sure the build configuration is checked out.
- 3 Expand the build configuration until the Build Targets entry is visible in the project tree. This will also cause the Sources section to appear in the content area.
- 4 Select the Build Targets heading or the name of the build configuration itself. Detail information for the build targets appears in the content area.
- 5 Click the Sources title to make it active.
- 6 Place a check mark next to the source or sources you want to delete.
- 7 Click the Delete icon (shaped like a red X). A confirmation dialog box asks you to confirm the deletion.
- 8 Click Yes to confirm the deletion; click Cancel to leave the sources as is.

Build Targets

About Build Targets

A build target is the desired end product of the build job, such as an executable file. An executable file is an example of a **real** build target.

Build targets can also be **virtual**. A virtual target is a target that represents an abstract idea, such as "all". A target of "all" would mean to build all targets, not an executable file named "all".

Each target is associated with a build configuration. You must create a build configuration before attempting to create a target. Remember to also have the build configuration checked out before attempting to modify it.

Creating Build Targets

When you create a new build target, you invoke the Create New Target Wizard. This wizard has four pages:

- Target page
Contains information about the target itself.
- Script page
Contains information about the script used to build the target.
- Options page
Contains information about build options and build option groups, if any.
- Build Plan page

Offers a chance to add a source file, attach an existing source file, define or attach dependent targets, or to go back and alter details on any of the three previous pages.

The remainder of this section describes the process of creating a new target, but does not go into detail for the Script and Options pages. Those pages are described in the [Scripts](#) section and the [Build Options](#) section.

To create a build target:

- 1** From the Dimensions administration console, select the build configuration that will receive the new target. The Build Targets section appears on the right side of the application window.
- 2** Click the New object icon (shaped like a plus sign). The Create New Target wizard appears. By default, the Target page appears first.
- 3** Fill out the Target page and the other three wizard pages as described in the following sections.

Fill out the Target page

On this page, you enter the identification details for the target.

- 1** For Name, enter a name for the target.
- 2** For File, enter the filename for the target. This must be the exact filename of the target on the build node.
- 3** For Relative Path, enter the path relative to the build area.
- 4** For Description, enter a description for the target.
- 5** Select the Is Virtual checkbox if the target is a virtual target.
- 6** Select the Is Final checkbox if the target is not an intermediate target; that is, the target will not be used as the input to a subsequent build operation.
- 7** Click Next to reach the Scripts page.

Fill out the Scripts page

On this page, you select the source for the script, and enter the text of the script if needed.

- 1** For **Select the source for script content when building the target**, choose from one of the following four choices:
 - **input script content manually**
 - **Use a Dimensions-controlled file as a script**
 - **Use a file in the build area as a script**
 - **Use the build configuration main script**

Enter the text for the script, or select one of the other three choices that designate a script that has already been created.

For a detailed description of all four choices, see the [Scripts](#) section on page [210](#).

- 2** Click Next to reach the Options page.

Fill out the Options page

On this page, you define build options, or attach previously-defined option groups.

Build options and option groups are not required, but can be attached here if desired.

- If you do not want to define an option or attach an option group, be sure Done is selected, then click Next to reach the Build Plan page.
- If you want to define a new build option:
 - a Select **Add new option** and then click Next. The Add New Option page appears.
 - b Fill out the Add New Option page as described in [Creating Build Options on page 219](#).
 - c Click Next. You are returned to the Options tab.
- If you want to attach a build option group:
 - a Click the Option Groups tab on the Options page.
 - b Select **Add new option group** and then click Next. The Add Option Group page appears.
 - c Select the desired option group and then click Next. You are returned to the Option Groups tab.

For a detailed description of options, see the [Build Options](#) section. For a discussion of option groups, see [Build Option Groups](#) on page 163.

When you are done with the Options page, click Next to reach the Build Plan page.

Fill out the Build Plan page

On this page, you define the inputs to the target. Inputs can be source files or other targets, either existing or yet to be defined.

You can also go back and modify an existing target or source file.

- If you want to add a source file:
 - a Select **Add new source** and Click Next. The Add New Source page appears.
 - b Enter the filename or mask (example: *.java).
 - c Enter the path to the file or files, relative to the build area. For the current directory, use ".\".
 - d Click Next. You are returned to the Build Plan page.
- If you want to add a target that the displayed target uses as an input:
 - a Select **Add new target**, then click Next. The Add New target page appears.
 - b Fill out the fields as described in [Fill out the Target page](#), then click Next. The Modify Script page appears.
 - c Fill out the Modify Script page as described in [Fill out the Scripts page](#), then click Next. The Modify Options page appears.
 - d Fill out the Modify Options page as described in [Fill out the Options page](#), then click Next. You are returned to the Build Plan page.

When you are done adding source files or targets, click Finish. The new targets and source files appear in the Build Targets area of the Dimensions administration console.

Modifying Build Targets

To modify an existing build target:

- 1 Make sure the build configuration containing the target is checked out.
- 2 Expand the build configuration until the target is visible in the project tree.
- 3 Select the name of the target. The target details appear in the content area.
- 4 Edit the desired target details.
- 5 Check the build configuration back in.

Deleting Build Targets

To delete an existing build target or targets:

- 1 Locate the build configuration containing the target you want to delete.
- 2 Make sure the build configuration is checked out.
- 3 Expand the build configuration until the Build Targets entry is visible in the project tree.
- 4 Select the Build Targets heading or the name of the build configuration itself. Detail information for the build targets appears in the content area.
- 5 Place a check mark next to the target or targets you want to delete.
- 6 Click the Delete icon (shaped like a red X). A confirmation dialog box asks you to confirm the deletion.
- 7 Click Yes to confirm the deletion; click Cancel to leave the targets as is.

Inputs

About Inputs

Inputs to a target can be sources, or other targets.

You can define new sources and targets as inputs, or select inputs from existing sources and targets.

Adding a New Target as an Input

To add a new target as an input:

- 1 In the Build Management tab, expand the navigation tree until you can see the target whose script you want to view.
- 2 Select the target. The Inputs section appears in the content area.
- 3 Click the **Add new target as an input** icon:



The Create New Target wizard appears.

- 4 Complete the Create New Target Wizard as described in [Creating Build Targets](#) on page 202.
- 5 When you are done with the wizard, check in the modified build configuration.

Adding a New Source as an Input

To add a new source as an input:

- 1 In the Build Management tab, expand the navigation tree until you can see the target whose script you want to view.
- 2 Select the target. The Inputs section appears in the content area.
- 3 Click the **Add new source as an input** icon:



The Add New Source wizard appears.

- 4 Enter a name for the new source.
- 5 Enter a path to the source that is relative to the project relative path. This restricts the path to the sub-directory that you specify. For example, if the project relative path is C:\Projects\Qlarius\Java and you specify a relative path of Utilities, the path to the source will be restricted to C:\Projects\Qlarius\Java\Utilities.

- 6 Click OK to accept the dialog box.
- 7 Check in the modified build configuration.

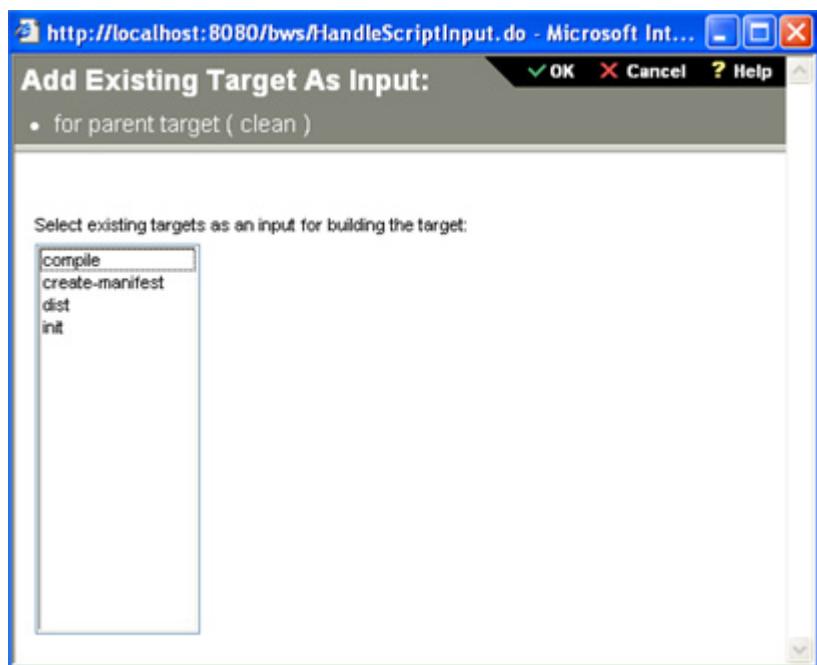
Adding an Existing Target as an Input

To add an existing target as an input:

- 1 In the Build Management tab, expand the navigation tree until you can see the target whose script you want to view.
- 2 Select the target. The Inputs section appears in the content area.
- 3 Click the **Add existing target as an input** icon:



The Add Existing Target As Input dialog box appears.



- 4 Select from the list of existing targets.
- 5 Click OK to accept the dialog box.
- 6 Check in the modified build configuration.

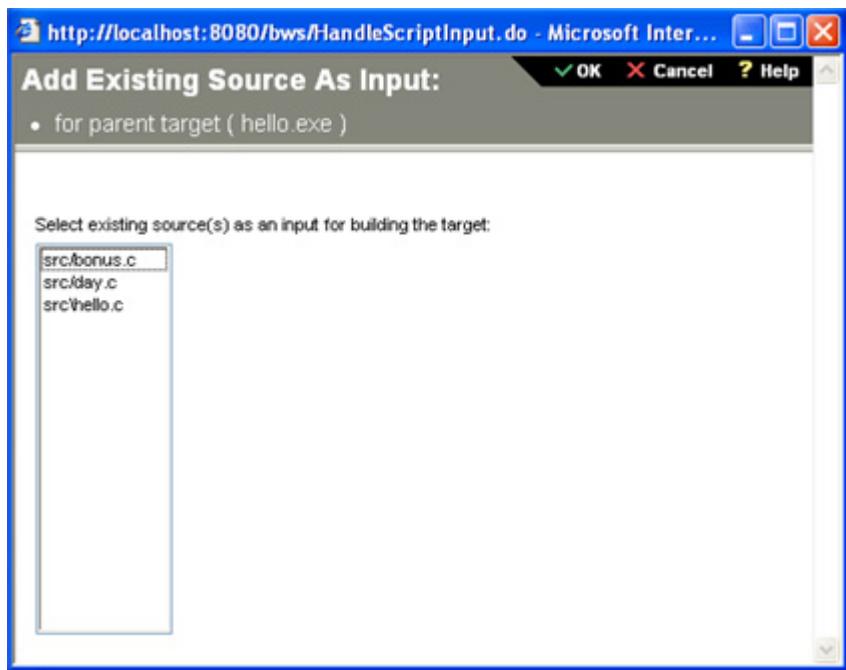
Adding an Existing Source as an Input

To add an existing source as an input:

- 1 In the Build Management tab, expand the navigation tree until you can see the target whose script you want to view.
- 2 Select the target. The Inputs section appears in the content area.
- 3 Click the **Add existing source as an input** icon:



The Add Existing Source As Input dialog box appears.



- 4 Select from the list of existing sources.
- 5 Click OK to accept the dialog box.
- 6 Check in the modified build configuration.

Modifying Inputs

To modify an input:

- 1 In the Build Management tab, expand the navigation tree until you can see the target whose input you want to modify.
- 2 Be sure that the build configuration is checked out.
- 3 Select the target. The Inputs section appears in the content area.

- 4 Select the input you wish to modify.
- 5 Click the Edit selected objects icon (shaped like a pencil). If the input is a source, the Edit Source dialog box appears. If the input is a target, the Edit Build Target dialog box appears.
- 6 Edit the input:
 - If the Edit Source dialog box appears, follow the instructions given in [Modifying Build Sources](#) on page 201.
 - If the Edit Build Target dialog box appears, follow the instructions given in [Modifying Build Targets](#) on page 205.
- 7 When you have finished modifying the input, check in the build configuration.

Deleting Inputs

To delete an input:

- 1 In the Build Management tab, expand the navigation tree until you can see the target whose input you want to delete.
- 2 Be sure that the build configuration is checked out.
- 3 Select the target. The Inputs section appears in the content area.
- 4 Select the input you wish to delete.
- 5 Click the Delete selected objects icon (shaped like a red X). A confirmation dialog box appears.
- 6 Click Yes to delete the input; click Cancel to leave the input as is.
- 7 When you have finished deleting the input, check in the build configuration.

Outputs

About Outputs

Build targets, both real and virtual, appear in the content area as outputs.

When you add a new output, you are essentially adding a new target. Although there are two icons next to the Outputs title—one for adding a new output, the other for editing the selected output—the icons call the Create New Target Wizard and Edit Build Target dialog box respectively.

Use the procedures described in [Creating Build Targets](#) on page 202 and [Modifying Build Targets](#) on page 205 to create and modify outputs.

To view the outputs list, select a specific target in the navigation tree.

Scripts

About Scripts

Scripts determine the specific behavior of a build job. For example, you can define a script to do any of the following:

- compile and link
- clean up the build area before the build executes
- rename files that have been transferred into the build area, before the build executes
- delete temporary files after the build executes

Scripts can be defined at a build configuration level or a target level. (Target-level scripts are known as transition scripts.) There are important differences between the two.

Build Configuration Scripts

Scripts associated with build configurations generally apply to the build as a whole, and come in four different varieties.

- Pre-Script

This type of script is executed before the main script, but after sources are transferred into the build area.

- Main Script

This script is not required, but can be useful when you have many targets that can use the same script. In that event, you can specify that individual targets should use this script. If the scripts are different, the transition script for an individual target will take precedence.

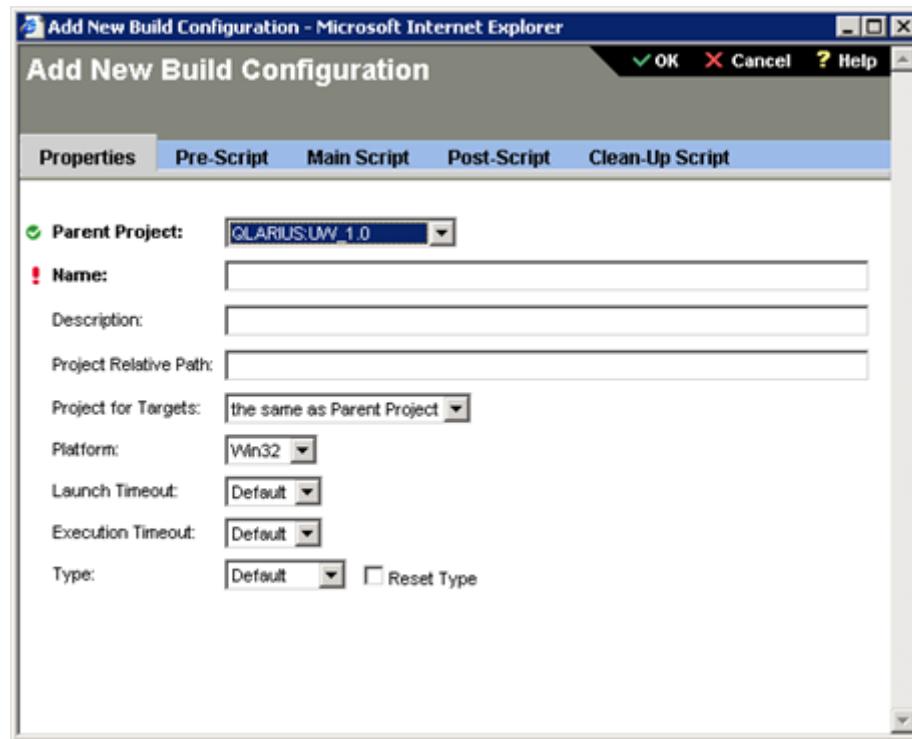
- Post-Script

This type of script is executed after the main script.

- Clean-Up Script

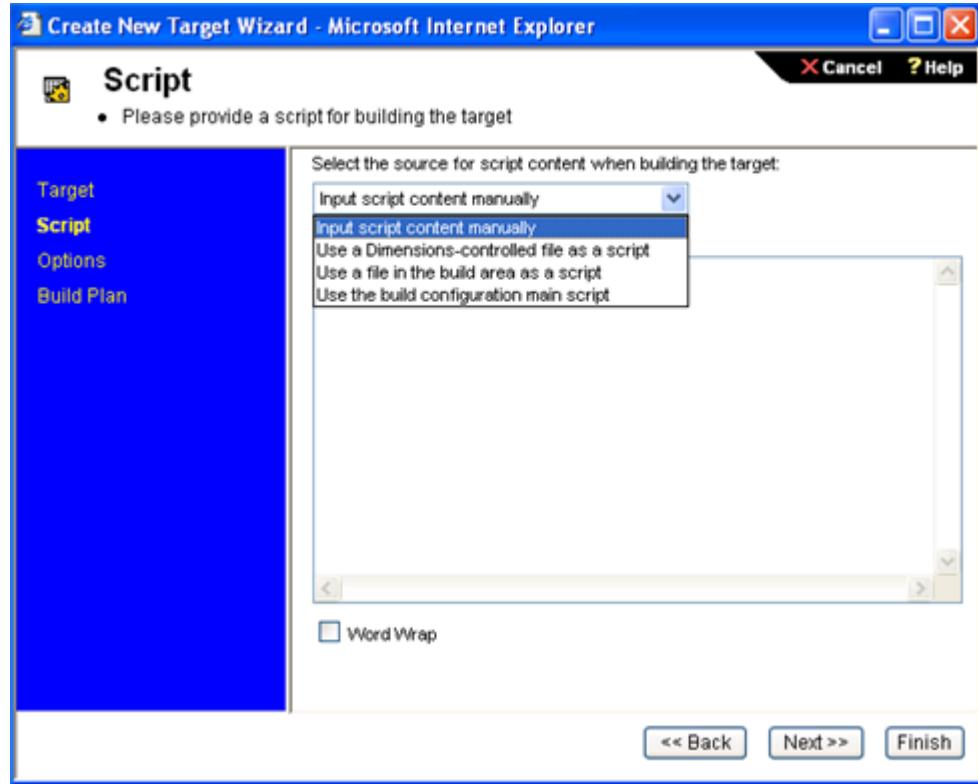
This type of script is executed before the main script, and before sources are transferred into the build area.

When you create a new build configuration, there is a tab for each of these scripts:



Transition Scripts

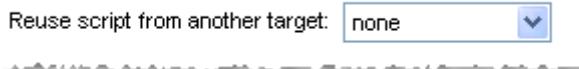
Transition scripts apply to one or more targets. Unlike build configuration scripts, transition scripts come in only one variety, and are defined at the time a target is created:



Reusing Scripts

Dimensions offers you two ways to reuse scripts.

- Once a transition script has been defined, any other target in the build configuration can use the script. The Target page of the Create New Target Wizard contains a field named **Reuse script from another target**:



The pulldown list allows you to select a script that has already been written for another target in the build configuration.

- You can also re-use a build configuration script as a transition script. Select **Use the build configuration main script** when creating a transition script.

Adding a Script

Whether you are creating a build configuration script or a transition script, you can select any of the following three ways to add a script:

- Input script content manually
- Use a Dimensions-controlled file as a script
- Use a file in the build area as a script

You can use each of these methods whether you are creating a Pre-Script, Main Script, or Post-Script, or Clean-Up Script.



NOTE When you create a new target, you must fill out at least the name of the target before you can move to the Script page of the Create New Target Wizard.

If you are creating a transition script, you can also select **Use the build configuration main script**.

Entering Content Manually

You can enter the content of the script manually. This method is best when you are creating test configurations, or when the content of the build script is simple enough so that you can type it from memory.

To enter the content of the script manually:

- 1 From the Script page of the Create New Target wizard, select **Input script content manually** as a script from the drop-down list. The Script Content text field appears.
- 2 Enter the content of the script into the Script Content area.



NOTE Dimensions will look for the build script and any other files in the build directory. If you want to refer to files that are elsewhere, you will have to specify the location of those files in the script.

- 3 Click Next to define build options, or click Finish to complete the target definition.

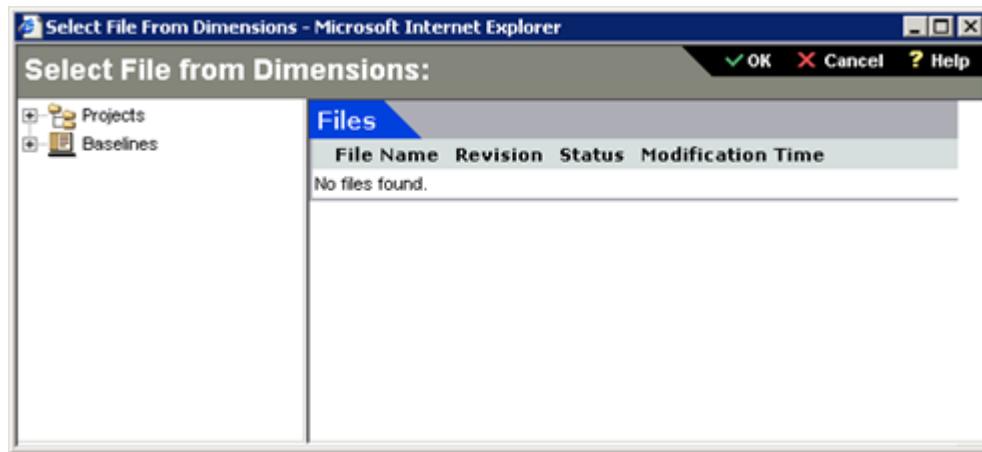
Using a Dimensions-Controlled File



NOTE If you select a Dimensions-controlled file as a script, be aware that the file will be pushed to the build area if you are using a work area. If you are using a deployment area, Dimensions Build assumes that the script has already been deployed to that area.

To use a Dimensions-controlled file as a script:

- 1 From the Script page of the Create New Target wizard, select **Use Dimensions-controlled file** as a script from the drop-down list. A text field appears, with a Browse button next to it.
- 2 Click the Browse button. The Select File from Dimensions dialog box appears:



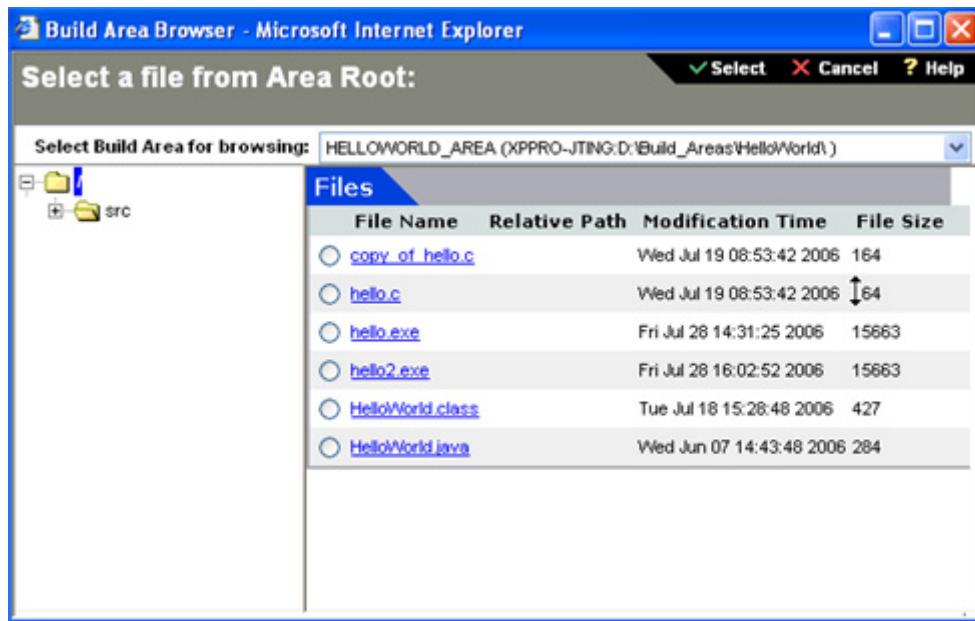
- 3 Expand the Projects or Baselines tree to display the available files.
- 4 Select the Dimensions-controlled file. A check mark appears next to the file name.
- 5 Click OK. The specification for the file appears in the text field for the script name.
- 6 Click Next to define build options, or click Finish to complete the target definition.

Using a File in the Build Area

To use a file in the build area as a script:

- 1 From the Script page of the Create New Target wizard, select **Use a file in the build area as a script** from the drop-down list. A text field appears.

- 2** If you already know the name of the file containing the build script, enter it in the text field of the dialog box; otherwise, click Browse to display the Build Area Browser:



- 3** Select the file from the browser, then click Select. The file appears in the Edit Transition Script dialog box.
- 4** Click Next to define build options, or click Finish to complete the target definition.

Using the Build Configuration Global Script

To use the build configuration global script:

- 1** From the Script page of the Create New Target wizard, select **Same as build configuration global script** from the pulldown list.
- 2** Click Next to define build options, or click Finish to complete the target definition.

Viewing a Build Configuration Script

To view a build configuration script:

- 1** In the Build Management tab, expand the navigation tree until you can see the build configuration whose script you want to view.
- 2** Select the build configuration.
- 3** Click the View icon or the Edit/View icon:
- If you have the build configuration checked out, the icon will be shaped like a pencil. Clicking the icon displays the Edit Build Configuration dialog box.
 - If the build configuration is not checked out, the icon will be shaped like a magnifying glass. Clicking the icon displays the View Build Configuration dialog box.

Viewing a Transition Script

To view a transition script:

- 1** In the Build Management tab, expand the navigation tree until you can see the target whose script you want to view.
- 2** Select the target. The Transition Details section appears in the content area.
- 3** Click the View icon or the Edit/View icon:
 - a** If you have the build configuration checked out, the icon next to the Transition Details title will be shaped like a pencil. Clicking the icon displays the Edit Transition Script dialog box.
 - b** If the build configuration is not checked out, the icon next to the Transition Details title will be shaped like a magnifying glass. Clicking the icon displays the View Transition Script dialog box.

Modifying a Build Configuration Script

The following procedure applies to Pre-Scripts, Main Scripts, Post-Scripts, and Clean-Up Scripts.

To modify a build configuration script:

- 1** If you already have the build configuration checked out and the Pre/Main/Post/Clean-Up Script tabs are visible, skip to [Step 9](#).
- 2** In the Dimensions Administration Console, click the Build Management tab.
- 3** Expand the Dimensions Projects tree until you can see the build configuration whose script you want to modify.
- 4** Select the project containing the build configuration. The Build Configurations section appears in the content area.
- 5** Place a check mark next to the build configuration that will receive the script.
- 6** Check out the build configuration by clicking the Check Out icon.
- 7** Click the pencil icon. The Edit Build Configuration dialog box appears.
- 8** Click the Pre/Main/Post/Clean-Up Script tab.
- 9** To modify the script, select one of the following three choices:
 - a** input script content manually
If you select this choice, enter or modify the script in the **Script Content** text box. Select the Word Wrap check box if you want to have the text wrap.
 - b** use a Dimensions-controlled file as a script. If you select this choice, enter or modify the location of the file, or click **Browse** to browse for it.



NOTE If you select a Dimensions-controlled file as a script, be aware that the file will be pushed to the build area if you are using a work area. If you are using a deployment area, Dimensions Build assumes that the script has already been deployed to that area.

- c use a file in the build area as a script

If you select this choice, enter or modify the location of the file, or click **Browse** to browse for it.
- 10** If you added or selected a new script, and need to delete an existing script, do so now.
- 11** Click OK to accept the dialog box.
- 12** Check in the modified build configuration by clicking the Check In icon.

Modifying a Transition Script

The following procedure applies to transition scripts.

To modify a transition script:

- 1** In the Dimensions Administration Console, click the Build Management tab.
- 2** Check out the build configuration containing the target whose transition script you wish to modify.
- 3** In the navigation tree, Select the target. The Transition Details section appears in the content area.
- 4** Click the pencil icon next to the Transition Details title. The Edit Transition Script dialog box appears.
- 5** If you want to select a different build tool, use the pulldown list to make a new selection for that field.
- 6** To modify the main part of the script, select one of the following four choices:
 - a input script content manually

If you select this choice, enter or modify the script in the **Script Content** text box. Select the Word Wrap check box if you want to have the text wrap.
 - b use a Dimensions-controlled file as a script

If you select this choice, enter or modify the location of the file, or click **Browse** to browse for it.



NOTE If you select a Dimensions-controlled file as a script, be aware that the file will be pushed to the build area if you are using a work area. If you are using a deployment area, Dimensions Build assumes that the script has already been deployed to that area.

- c use a file in the build area as a script

If you select this choice, enter or modify the location of the file, or click **Browse** to browse for it.
 - d use the build configuration main script

If you select this choice, you need not enter any other information, as Dimensions knows where to find the main script.
- 7** If you added or selected a new script, and need to delete an existing script, do so now.

- 8 Click OK to accept the dialog box.
- 9 Check in the modified build configuration by clicking the Check In icon.

Deleting a Build Configuration Script

Use the following procedure to delete a build configuration script. Remember that at least the main script must be present for a build configuration to run successfully.

To delete a build configuration script:

- 1 If you already have the build configuration checked out and the Pre/Main/Post-Script tab is visible, skip to [Step 9](#).
- 2 In the Dimensions Administration Console, click the Build Management tab.
- 3 Expand the Dimensions Projects tree until you can see the build configuration whose script you want to delete.
- 4 Select the project containing the build configuration. The Build Configurations tab appears.
- 5 Place a check mark next to the build configuration to be modified.
- 6 Check out the build configuration by clicking the Check Out icon.
- 7 Click the pencil icon. The Edit Build Configuration dialog box appears.
- 8 Click the Pre/Main/Post-Script tab.
- 9 Select **input script content manually** if it is not selected already.
- 10 Delete any text that appears in the **Script Content** text box.
- 11 Click OK to accept the dialog box.
- 12 Check in the modified build configuration by clicking the Check In icon.

Deleting a Transition Script

Use the following procedure to delete a transition script.

To delete a transition script:

- 1 In the Dimensions Administration Console, click the Build Management tab.
- 2 Check out the build configuration containing the target whose transition script you wish to delete.
- 3 In the navigation tree, select the target. The Transition Details section appears in the content area.
- 4 Click the pencil icon next to the Transition Details title. The Edit Transition Script dialog box appears.
- 5 Delete any text that appears in the **Script Content** text box. (You should of course replace the deleted text using some other source, or the target may not build successfully.)

- 6 Click OK to accept the dialog box.
- 7 Check in the modified build configuration by clicking the Check In icon.

Build Options

About Build Options

Build options provide a way of passing configuration information to the build environment. You can define a build option to pass along compiler flags, or you can define a build option to specify the path to a specific compiler. For details of the build options see *Dimensions Build Predefined Symbols* in *The Templating Language and Processor* chapter of the *Developer's Reference*.

Dimensions offers you several different ways to define build options.

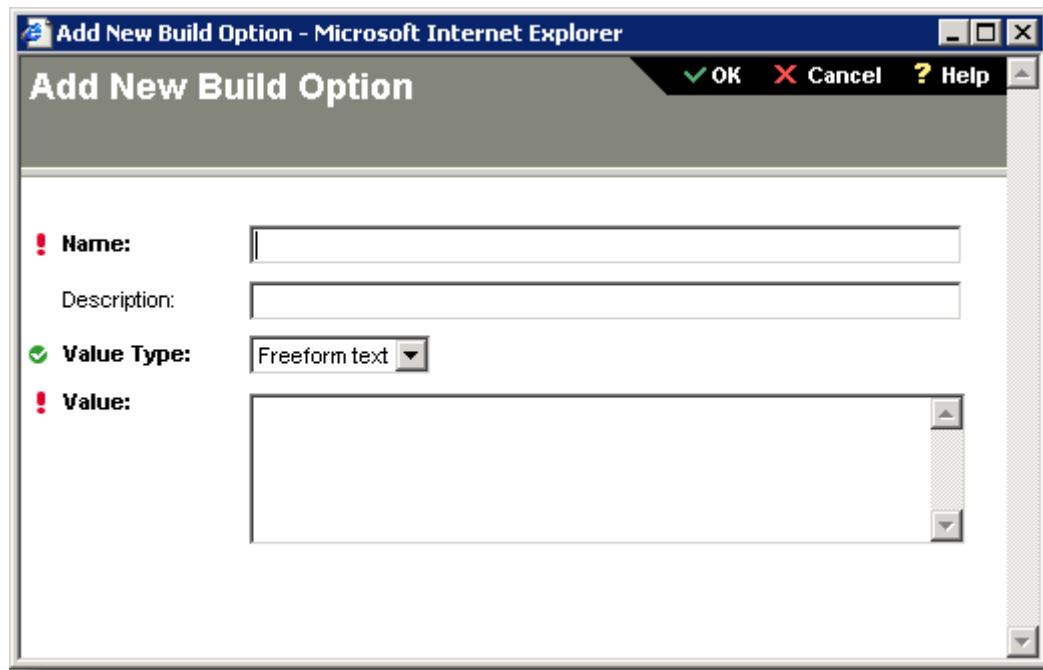
- From the Build Configuration
Define options at the build configuration level if you want the options to be available to all targets in that build, as well as to the Pre-Script, Post-Script, and Clean-Up Script. An example of this would be a particular search path.
- From the Build Area
Define options at the build area level if you want the options to be available to all builds in that area. An example of this would be the name of the License Manager host.
- From the Target
Define options at the target level when they are specific to a particular target. An example of this would be a debug or trace option.
Options defined during the process of creating a new target, or editing an old target, are actually defined at the transition level. Therefore, the options are available to any other target in the build configuration. In fact, you can easily attach the options defined for one target to a second target by selecting **Reuse script from another target** when you create the second target.
- From the Run Build Wizard
Define options in the Run Build wizard if you want the options to apply only to that one execution of the build. An example of this would be to add an option to print the time of execution.

Creating Build Options

To create a build option from a build configuration, build area, or target:

- 1 Check out the build configuration if you have not already done so.
- 2 Select the build configuration, build area, or target.

- 3 Click the New Object icon (shaped like a plus sign) next to the Build Options area. The Add New Build Option dialog box appears.



- 4 Enter a **Name** for the option. This is the name that will be used in build scripts later. The name can contain spaces.



NOTE If you use the same name for build options in all areas, the following precedence table applies, in order of highest priority to lowest priority:

- 1 Build options defined in the Run Build wizard
- 2 Build options defined at the target level
- 3 Build options defined at the build area level
- 4 Build options defined at the build configuration level

- 5 Enter an optional **Description** for the option.
- 6 For the **Value Type**, select from Freeform text, Option Rule, and Option Link.
 - Freeform text
 - Option Rule
 - Option Link
- 7 Enter the **Value** for the option.
- 8 Click **OK**. A confirmation dialog box appears briefly, and the new option appears in the Build Options area.

To define a build option from the Run Build wizard:

- 1** From the Run Build wizard, click Options.
- 2** Click Add New Option, then click Next. The **Add New Re-defined Option Value** page appears.
- 3** Enter a Name for the option.
- 4** Enter a Value for the option.
- 5** Click Next. The new option appears in the list of options on the Options page of the wizard.



NOTE A build option that is created or redefined in the Run Build wizard will last only as long as the Run Build wizard itself, or until the build job is run. If you close and reopen the Run Build wizard, the option definition will not be saved.

Modifying Build Options

You cannot edit build options defined in the Run Build wizard, because they are not saved with the build job. You can edit build options defined in a build configuration, build area, or build target.

To edit a build option:

- 1** Locate the build option you want to modify.
- 2** Make sure the build configuration containing the build option is checked out.
- 3** Select the build area, build configuration, or target. The Build Options section appears on the right side of the screen.
- 4** Click the name of the build option. The Edit Build Option dialog box appears.
- 5** Make the desired edits to the build option, then click OK. A brief confirmation dialog appears, and the edited build option appears in the Build Options section.

Deleting Build Options

To delete a build option:

- 1** Locate the build configuration containing the build option you want to delete.
- 2** Make sure the build configuration is checked out.
- 3** Select the entry containing the build option:
 - a** If the build option is attached to the build configuration, select the build configuration name. The Build Options heading should be visible in the content area.
 - b** If the build option is attached to a specific target, select the target name. The Build Options heading should be visible in the content area.
- 4** Place a check mark next to the option or options you want to delete.

- 5 Click the Delete icon (shaped like a red X). A confirmation dialog box asks you to confirm the deletion.
- 6 Click Yes to confirm the deletion; click Cancel to leave the build option as is.

Creating Build Option Groups

See [Build Option Groups](#) on page 163 for instructions on creating build options groups.

Versions

About Versions

Versions represent previously-checked-in editions of a build configuration. You can do only certain things with previous versions:

- You can view previous versions of a build configuration, as described in [Viewing Build Configurations](#) on page 186.
- You can copy previous versions of a build configuration, as described in [Copying Build Configurations](#) on page 189.
- You can rollback to a previous version.
- You can rewrite the current build configuration so that it is identical to a previous version.
- You cannot delete a previous version.

For information on how to rollback or rewrite, see [Versioning](#) on page 243.

Exporting and Importing

About Exporting and Importing

You can export your Dimensions build configurations to an external XML file. These files can be imported at a later time, or to a different machine.

You can also import build configuration files from either Ant or the Openmake/Serena ChangeMan Builder products. The procedures for importing build configurations from either of those products differ from the procedures presented in [Importing a Build Configuration](#) on page 223. See [Using Dimensions with Ant](#), on page 293, and [Using Dimensions Build with Openmake](#), on page 301.

Exporting a Build Configuration

If you need to save your build configuration, you can export the build configuration to an XML file.

To export a build configuration to an XML file:

- 1 Select the name of the surrounding project.
- 2 Select the name of the build configurations you want to export.
- 3 Click the Export icon.
- 4 A browser window opens with the XML file.
- 5 Save the XML file.



NOTE You can use the default filename of HandleBuildConfig.do, or rename the file as you see fit.

Importing a Build Configuration

Once you have saved a build configuration to an XML file, you can restore that build configuration by importing the XML file.

Before You Begin

- It's best if you have a project ready to receive the imported build configuration.
- Don't worry about the configuration having the same name as the one from which it was exported; you will be given an opportunity to rename it during the import.

To import a build configuration:

- 1 Select the name of the project that will receive the build configuration.
- 2 In the Build Configurations section, click the Import build configuration icon. The Import Build Configuration Wizard appears.
- 3 Enter or browse for the name of the XML file to be imported.
- 4 Select the name of the build configuration you want to import. (If there is only one build configuration in the file, the wizard skips this step.)
- 5 Enter a new name for the build configuration.
- 6 Click Finish to confirm the import operation.
- 7 A brief confirmation displays, and the new build configuration appears in the Build Configurations list.

Importing an Ant build.xml File

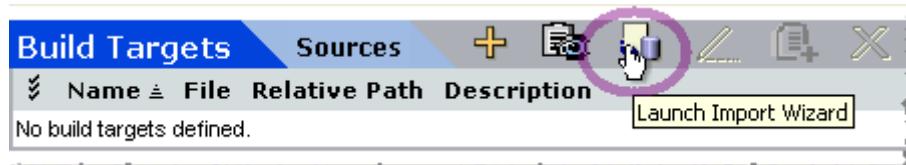
Users of Ant can import their build.xml files into build configurations in Dimensions.



NOTE Importing Ant and Openmake files requires the use the Import Wizard—not the Import icon that appears on the top-level row of icons when a project or the Dimensions Projects heading is selected.

To import an Ant build.xml file into Dimensions:

- 1 In Dimensions, select the Build Management tab.
- 2 In the navigation area, click Build Projects, expand your build project, and click the build configuration where you want to import the build.xml file.
- 3 In the content area, in the Build Targets section, click Launch Import Wizard.



The Import Wizard appears. Note that on the left side of the wizard there are links to four pages: Select Import Type, Choose File, Parameters, and Confirm. By default, the Select Import Type page is displayed.

- 4 On the Select Import Type page, select **Ant build configuration (build.xml) file** and click **Next**.
- 5 On the Choose File page, in the **Select Ant build configuration file** field, do one of the following:
 - Type the full path to the build.xml file that you want to import.
 - Click **Browse**, navigate to the build.xml file that you want to import, select it, and click **Open**.
 Click **Next**.
- 6 On the **Parameters** page, select the sources that you want to import from the build.xml file, then click **Next**.
- 7 On the **Confirm** page review the details of the target that you are going to create and click **Finish**. Dimensions briefly displays a confirmation dialog, and afterwards you can see the new targets in the Build Targets section.
- 8 Click the Sources tab. Verify that the new sources were added to your list of build sources.

- 9 To review the transition script that will be used to build the target do the following:
 - a In the navigation area, select the build target that you just imported.
 - b In the content areas, in the Transition Details section, click Edit Build Script.



The Edit Transition Script dialog box appears. The Script Content field displays the following default script:

```
ant -buildfile "build.xml" "<name of target>"
```

- c Modify the script if required.
- d Click OK.

- 10 Optionally add build options and option groups to the build target.

Importing an Openmake Build Configuration

Users of the Openmake product from Catalyst Systems Corporation, or of the Serena ChangeMan Builder product, can import the TGT file that defines build targets under either of those two systems.

See [Using Dimensions Build with Openmake](#) on page 301.

Tips

Here are a few build configuration tips.

- When you create a new build configuration, it is automatically checked out. You must check it in before running it.
- If you Undo Checkout on a build configuration that has not been checked in, all details will be lost without warning.
- When the build configuration traffic light symbol shows a yellow light, it means that the build is in progress.

Chapter 9

Executing Builds in the Build Administrator

About Executing Builds	228
Running a Single Build Configuration	228
Running All Build Configurations in a Project	230
Running a Build from a Scheduled Start	231
Running a Non-Scheduled Build from the Build Scheduling Tab	231
Running a Build Using Launch New Build	231
Rebuilding Previously Executed Build Jobs	232

About Executing Builds

You can execute builds from:

- the Dimensions Build administration console
- the Dimensions Web client
- the Dimensions desktop client
- the Dimensions ISPF on MVS client

This section describes the ways of executing builds in the Dimensions Administration Console.

For information on executing builds in the Web client and desktop client, see the *Serena Dimensions CM User's Guide*, and the help for those two clients.

For information on executing builds in the ISPF client, see the *Serena Dimensions for z/OS User's Guide*, and the ISPF panel help.

The possible ways of executing builds from the Dimensions Build administration console are:

- Running a single build configuration
- Running all build configurations in a project
- Running a build using a scheduled start
- Running a non-scheduled build from the Build Scheduling tab
- Using the **Launch new build** button from the Build execution statuses dialog box of an executed build
- Rebuilding a build from the execution history in the Build Monitoring tab

Running a Single Build Configuration

Follow this procedure to run a single build from the Dimensions Administration Console.

To run a single build configuration:

- 1 From the Build Management tab of the Dimensions Administration Console, expand the list of Dimensions Projects.
- 2 Select the project containing the desired build configuration.
- 3 Click the Run icon. The Build Page of the Run Build wizard appears.
- 4 To fill out the Build Page, select the desired values for each of the following fields:
 - Dimensions Project: The Dimensions project you selected earlier should automatically be displayed.
 - Baseline: If any part of the build is associated with a baseline, this field appears. Select the desired baseline.
 - Build Configuration: Select the desired build configuration.

- Version: Select **Latest** to build the latest version, or select a previously-executed version.
- Build Area: Select **All areas** to build in all build areas, or select a specific build area.
- Build Target: Select a specific target.

5 Click Next. The Build Scope page appears.

6 To fill out the Build Scope page, select the desired values for each of the following fields:

- Execute clean-up script before build: Check this box to have the clean-up script defined in the build configuration executed before the build begins. (If no clean-up script is defined, nothing will happen.)
- Do not transfer sources: Check this box to stop Dimensions from retrieving copies of the source files to the build area.
- Do not preserve targets: Check this box to stop Dimensions from preserving the target files.
- Select Dimensions Project for targets preservation: Select the Dimensions project to receive the target files produced by the build.

7 Click Next. The Change Requests page appears.

8 To fill out the Change Requests page, select the requests that you wish to associate with the build from the displayed list. If the desired request is not listed, you can enter it directly into the text field.

9 Click Next. The Options page appears.

10 Complete the Options page by selecting the desired values for each of the following fields:

- Done: Select this when you have no more changes to make to the options.
- Add new option: Select this when you want to add a new option. Follow the procedure described in [Creating Build Options](#) on page 219.
- Modify selected option: Select this when you want to modify the selected option. Follow the procedure described in [Modifying Build Options](#) on page 221.
- Remove selected option(s): Select this when you want to remove the selected option.

11 Click Run. When the build confirmation status appears, you can evaluate the success or failure of the build. The various icons that appear in the build execution status dialog box are described in [Monitoring Builds in the Build Administrator](#) on page 261.

Running All Build Configurations in a Project

Follow this procedure to run all build configurations in a project, using the Dimensions Administration Console.

To run all build configurations in a project:

- 1** From the Build Management tab of the Dimensions Administration Console, expand the list of Dimensions Projects.
- 2** Select the project containing the desired build configurations.
- 3** Click the Run icon. The Build Page of the Run Build wizard appears.
- 4** To fill out the Build Page, select the desired values for each of the following fields:
 - Dimensions Project: Select the Dimensions project that contains the desired build configuration.
 - Baseline: If any part of the build is associated with a baseline, this field appears. Select the desired baseline.
 - Build Configuration: Select **All configurations**.
 - Version: **Latest** is selected by default. It is not possible to select specific versions when you are building all configurations.
 - Build Area: **All areas** is selected by default. It is not possible to select specific build areas when you are building all configurations.
 - Build Target: **All targets** is selected by default.
- 5** Click Next. The Build Scope page appears.
- 6** To fill out the Build Scope page, select the desired values for each of the following fields:
 - Execute clean-up script before build: Check this box to have the clean-up script defined in the build configuration executed before the build begins. (If no clean-up script is defined, nothing will happen.)
 - Do not transfer sources: Check this box to stop Dimensions from retrieving copies of the source files to the build area.
 - Do not preserve targets: Check this box to stop Dimensions from preserving the target files.
 - Select Dimensions Project for targets preservation: Select the Dimensions project to receive the target files produced by the build.
- 7** Click Next. The Change Requests page appears.
- 8** To fill out the Change Requests page, select the requests that you wish to associate with the build from the displayed list. If the desired request is not listed, you can enter it directly into the text field.
- 9** Click Next. The Options page appears.

- 10** Complete the Options page by selecting the desired values for each of the following fields:
 - Done: Select this when you have no more changes to make to the options.
 - Add new option: Select this when you want to add a new option. Follow the procedure described in [Creating Build Options](#) on page 219.
 - Modify selected option: Select this when you want to modify the selected option. Follow the procedure described in [Modifying Build Options](#) on page 221.
 - Remove selected option(s): Select this when you want to remove the selected option.
- 11** Click Run. When the build execution status appears, you can evaluate the success or failure of the builds. The various icons that appear in the build execution status dialog box are described in [Monitoring Builds in the Build Administrator](#) on page 261.

Running a Build from a Scheduled Start

To run a build in a delayed or scheduled fashion requires that you create a schedule for the build. See [Scheduling Build Jobs](#) on page 249.

Running a Non-Scheduled Build from the Build Scheduling Tab

You can also run a build from the Build Scheduling tab, without having to schedule it.

- 1** Select the build you wish to run from the project tree.
- 2** Click Run. The Run Build wizard appears.
- 3** Fill out the Run Build wizard as described in [Running a Single Build Configuration](#) or [Running All Build Configurations in a Project](#).



NOTE If you select a specific version of a build, the build must have been executed at least once or you may get an error referring to build areas.

Running a Build Using Launch New Build

The Launch New Build button appears on the Build Execution Results dialog box. Clicking this button returns you to the Build Page of the Run Build wizard. From there, you can rerun the build as described in [Running a Single Build Configuration](#) or [Running All Build Configurations in a Project](#).

Rebuilding Previously Executed Build Jobs

Follow this procedure to rebuild a build configuration that has been previously executed, using the Build Monitoring tab.

To rebuild a previously built build configuration:

- 1 Click the Build Monitoring tab.
- 2 Expand the History tree. A list of Dimensions projects appears.
- 3 Expand the Dimensions project that contains the desired build configuration.
- 4 Select the desired build configuration. The Execution History tab appears, with a list of previously executed build jobs.

The screenshot shows the Build Monitoring tab selected in the top navigation bar. Below it is a toolbar with Refresh, Rebuild, and Filter buttons. The main area is divided into two sections: a tree view on the left and a table on the right.

Left Panel (Tree View):

- Running Jobs
- History
 - \$GENERIC:\$GLOBAL
 - PAYOUT:PRJ_DEV_REL
 - PAYOUT:PRJ_INITIAL
 - Ant_Import_Test
 - CopyTest
 - HelloWorld** (highlighted in blue)

Right Panel (Execution History):

ID	Status	Time Start	Time Stop
116		Fri Jul-28 3:57PM	Fri Jul-28 3:57PM
112		Fri Jul-28 2:31PM	Fri Jul-28 2:31PM
90		Thu Jul-27 10:00AM	Thu Jul-27 10:00AM

You can identify the previously executed build jobs by their ID number, the start time, or, if you expand the build configuration, by the version.

- 5 Check the box next to the build job you wish to rebuild. The Rebuild icon becomes enabled.

- 6 Click Rebuild. The Rebuild Build Job dialog box appears.



- 7 Select the build area to use from the list.
- 8 Select the desired values for each of the following fields:
- Execute clean-up script before build: Check this box to have the clean-up script defined in the build configuration executed before the build begins. If no clean-up script is defined, Dimensions will not perform any clean-up action.
 - Do not transfer sources: Check this box to stop Dimensions from retrieving copies of the source files to the build area.
 - Do not preserve targets: Check this box to stop Dimensions from preserving the target files.
 - Select Dimensions Project for targets preservation: Select the Dimensions project to receive the target files produced by the build.
- 9 Click OK. When the build confirmation status appears, you can evaluate the success or failure of the build. The various icons that appear in the build execution status dialog box are described in [Monitoring Builds in the Build Administrator](#) on page 261.

Chapter 10

Using Filters

About Using Filters	236
What Can Be Filtered	236
Filtering Build Jobs by User Name	236
Filtering Build Jobs by Date and Time	238
Filtering Dimensions Projects	240
Editing Filter Conditions	241

About Using Filters

This section describes how to use filters in Dimensions.

Filters define criteria that the data must match in order to be displayed. A photographic filter can limit the wavelengths of light visible to a camera; a filter in Dimensions can limit the build projects or build jobs visible to the user.

You can also use filters to specify which build projects or build jobs should be hidden, instead of visible. For example, you can use a filter to limit the display of past builds, or to exclude all projects other than the one you are working on.

These tasks are discussed in this chapter.

What Can Be Filtered

Dimensions allows you to filter either build projects, or build jobs.

- On the Build Management tab: Dimensions projects
- On the Build Scheduling tab: Dimensions projects
- On the Build Monitoring tab: Build jobs (running jobs and History)
- On the Notifications tab: Dimensions projects

Filtering Build Jobs by User Name

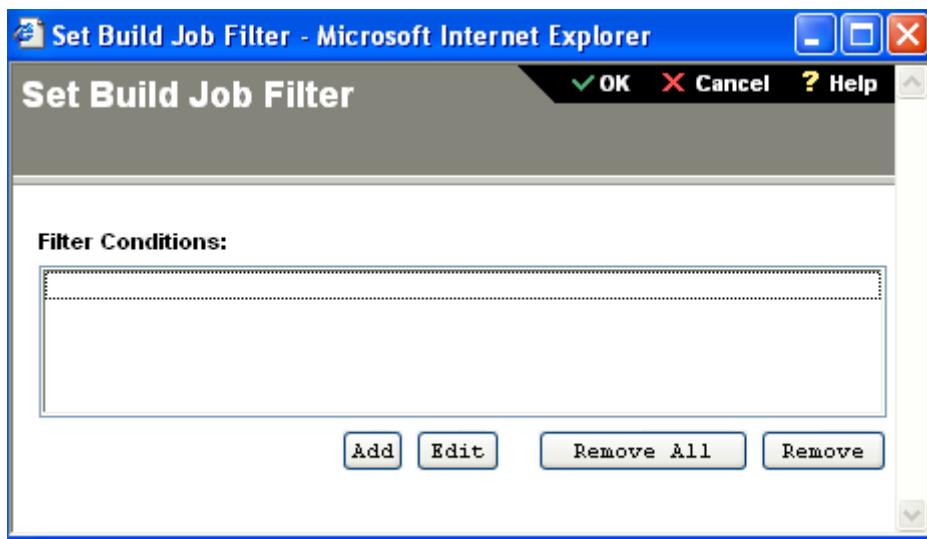
You can limit the displayed build jobs to a specific user name by setting a user name filter.

To set a build job user name filter:

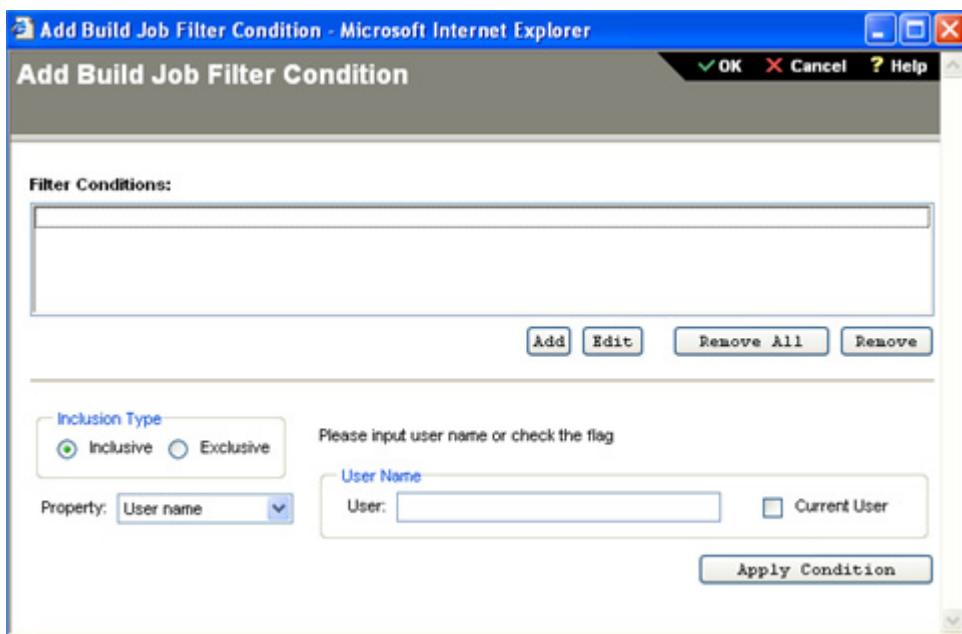
- 1 Select the Build Monitoring tab.
- 2 From the navigation tree in the left-hand part of the Dimensions screen, click Running Jobs. The Filter icon appears.



- 3 Click the Filter icon. The Set Build Job Filter dialog box appears.



- 4 Click Add. The Add Build Job Filter Condition dialog box appears.



- 5 Define the filter condition using the following guidelines:

- If you want to view your own build jobs, click the check box labeled Current User.
- If you want to view build jobs by another user, enter the name of that user.



NOTE The filter defines an exact match. Entering a string such as "Smith" will match "Smith" only—not "Smithson" or "Smithee". To match multiple strings, use the asterisk (*) as a wildcard—in this example, you would enter "Smith*".

- If you want to exclude build jobs by yourself or by another user, select Exclusive from the Inclusion Type field.

- To match (or exclude) multiple user names, enter multiple conditions. You can only specify one name per condition.
- 6 When you have finished defining the filter condition, click Apply Condition. The filter condition appears in the Filter Conditions list box.
 - 7 Click OK to accept the Add Build Job Filter Condition dialog box. The Active Jobs heading changes to indicate that the view is filtered.

Active Jobs (filtered)		
Status	Time Start	Build
<input type="checkbox"/>		Tue Aug-01 11:15AM
<input type="checkbox"/>		Tue Aug-01 11:10AM
<input type="checkbox"/>		Tue Aug-01 11:09AM

Filtering Build Jobs by Date and Time

You can limit the displayed build jobs to a specific date and time range by setting a range filter.

To set a build job date and time range filter:

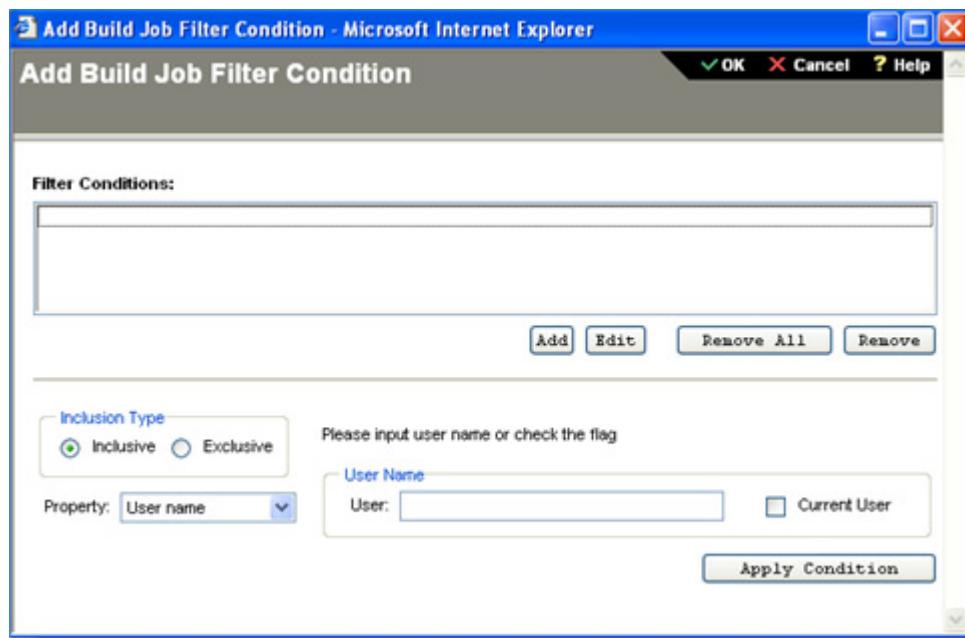
- 1 Select the Build Monitoring tab.
- 2 From the navigation tree in the left-hand part of the Dimensions screen, click the name of the project whose build jobs you wish to filter. The Filter icon appears.



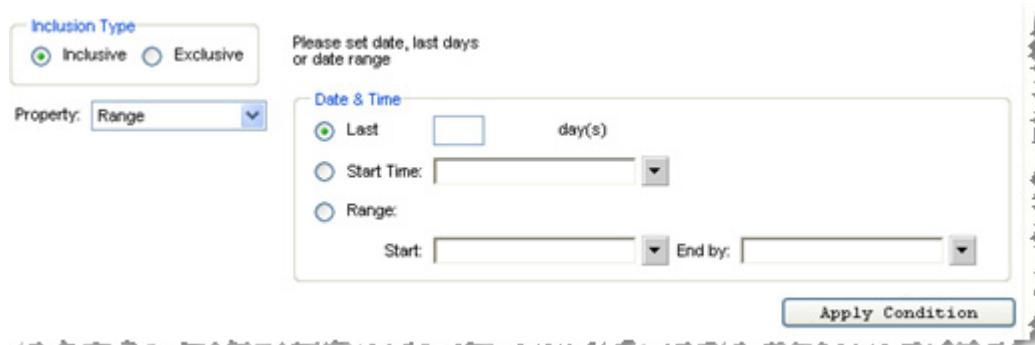
- 3 Click the Filter icon. The Set Build Job Filter dialog box appears.



- 4 Click Add. The Add Build Job Filter Condition dialog box appears.



- 5 In the Property field, select **Range**. The date and time fields appear.



- 6 Define the filter condition using the following guidelines:

- To match (or exclude) build jobs from the last N days, where N is an integer, enter a value in the Last day(s) field.
- To match (or exclude) build jobs since a certain Start Time, click the down arrow and select the date and time.
- To match (or exclude) build jobs within a certain range, click the down arrow next to Start and select a starting date and time, then click the down arrow to the right of End by and select an ending date and time.

- 7 When you have finished defining the filter condition, click **Apply Condition**. The filter condition appears in the Filter Conditions list box.
- 8 Click **OK** to accept the Add Build Job Filter Condition dialog box. The Active Jobs heading changes to indicate that the view is filtered.

Filtering Dimensions Projects

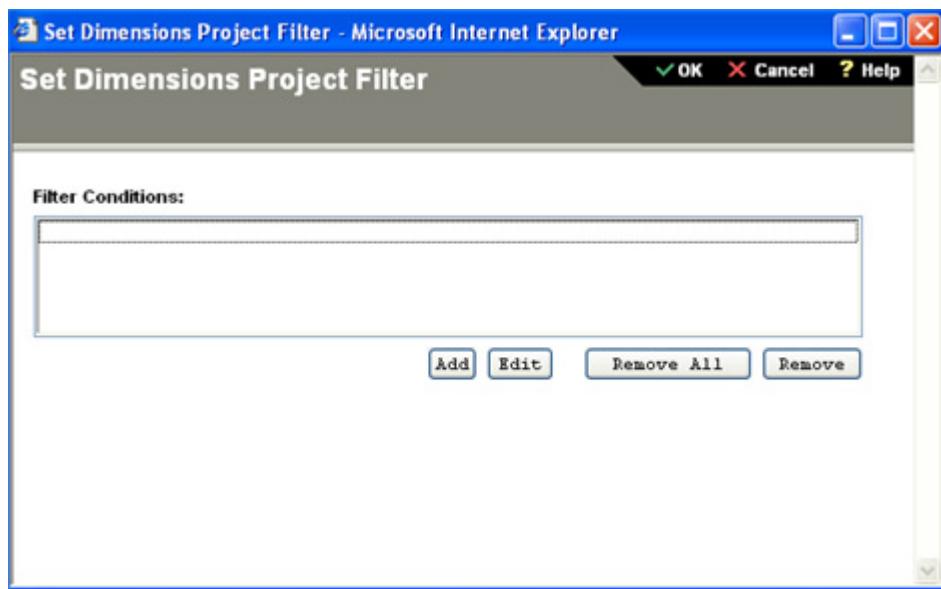
You can limit the displayed Dimensions projects with a filter.

To set a Dimensions project filter:

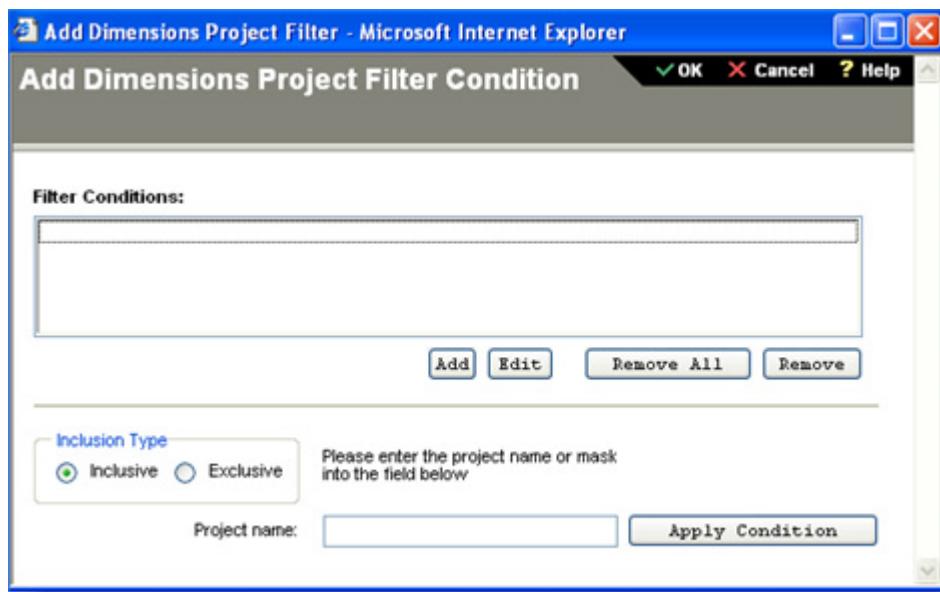
- 1 From the navigation tree in the left-hand part of the Dimensions screen, click the highest-level entry that uses filters. (For example, on the Notifications tab, click Subscriptions.) The Filter icon appears.



- 2 Click the Filter icon. The Set Dimensions Project Filter dialog box appears.



- 3 Click Add. The Add Dimensions Project Filter Condition dialog box appears.



- 4 Enter the name of the project you wish to search for (or exclude).



NOTE The filter defines an exact match. Entering a string such as "Project" will match "Project" only—not "Project1" or "New_Project". To match multiple strings, use the asterisk (*) as a wildcard—in this example, you would enter "*Project*".

- 5 If you wish to exclude the named project, click the Exclusive radio button.
- 6 When you have finished defining the filter condition, click Apply Condition. The filter condition appears in the Filter Conditions list box.
- 7 Click OK to accept the Add Dimensions Project Filter Condition dialog box. The Dimensions Projects heading changes to indicate that the view is filtered.

Dimensions Projects (filtered)	
Name	Description
QLARIUS:DM10	DM10
QLARIUS:PRJ_MVS	Project PRJ_MVS.AAAA

Editing Filter Conditions

Editing filter conditions includes any of the following:

- Altering the definition of the filter
- Removing specific filter conditions
- Removing all filter conditions

Altering the Definition of a Filter Condition

To alter the definition of a filter condition:

- 1 Click the Filter icon. The filter dialog box appears.
- 2 Select the specific filter condition you wish to alter. You can only select one condition at a time.
- 3 Make the desired changes to the filter condition, then click Apply Condition. The altered filter condition appears in the list of Filter Conditions.
- 4 Click OK to accept the dialog box.

Removing Specific Filter Conditions

To remove specific filter conditions:

- 1 Click the Filter icon. The filter dialog box appears.
- 2 Select the specific filter condition you wish to remove. You can only select one condition at a time.
- 3 Click Remove. Dimensions deletes the selected filter condition.
- 4 Repeat until all of the filter conditions you wish to delete are gone.
- 5 Click OK to accept the dialog box.

Removing All Filter Conditions

To remove all filter conditions:

- 1 Click the Filter icon. The filter dialog box appears.
- 2 Click Remove All. All of the filter conditions disappear.
- 3 Click OK to accept the dialog box.

Chapter 11

Versioning

About Versioning	244
Example of Rollback	244
Example of Rewrite	246
Deleting Versions Not Allowed	247

About Versioning

This section describes how to work with different versions of your build configurations.

The three main things you can do are:

- Rollback

If you select an existing version of a build configuration and then click the Rollback icon, Dimensions will create a new version identical to the selected version, and check it in with a new version number. Use this feature to return to a previous version.

- Rewrite

If you select an existing version of a build project and then click the Rewrite icon, Dimensions will rewrite the current build configuration so that it is identical to the selected build configuration. The current build configuration is NOT checked in. Use this feature to return to an existing build configuration so that you can create a variation on it.



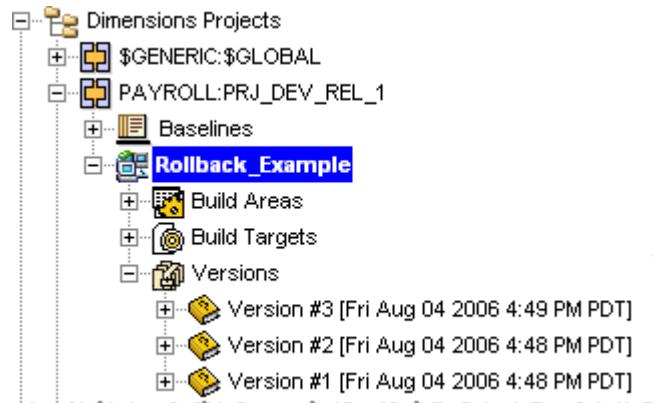
NOTE You can only Rollback or Rewrite from the Build Management tab. You cannot perform these operations from the other tabs.

- Copy a version and save it as another build configuration

If you select an existing version of a build configuration and then click the Copy icon, you can save the copied version under a new name. See [Copying Build Configurations](#) on page 189.

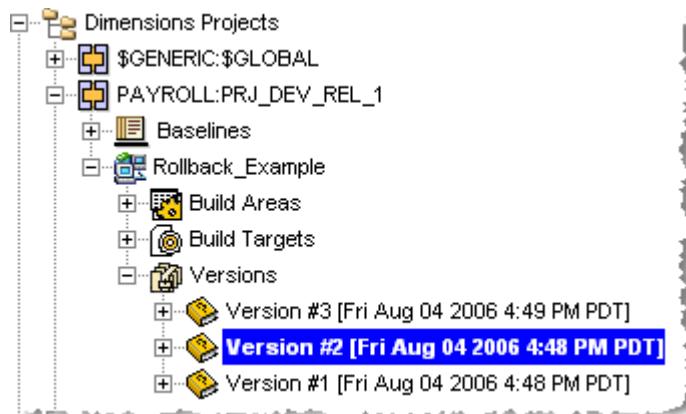
Example of Rollback

Consider a build configuration with several existing versions:



To roll back to a previous version:

- 1 Select the desired version (in this example, Version #2).



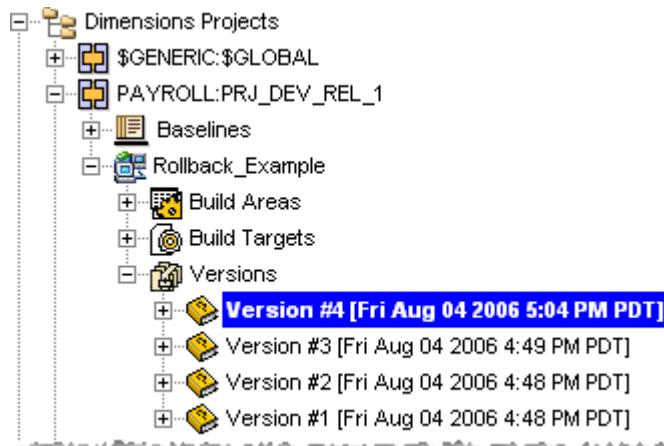
- 2 Click the Rollback icon.



The Rollback dialog box appears:

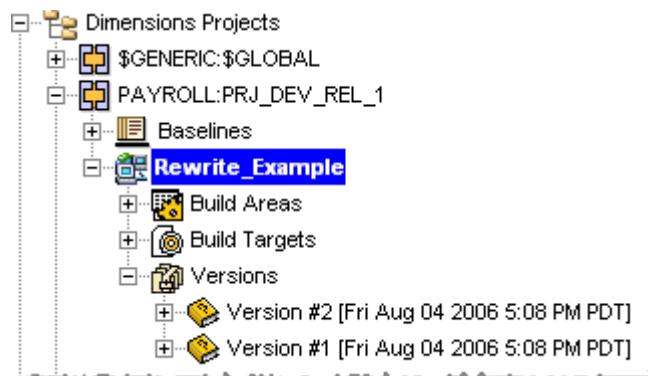


- 3 Enter a comment (as shown above), then click OK. A dialog box briefly confirms the rollback operation, and a new version appears in Dimensions:



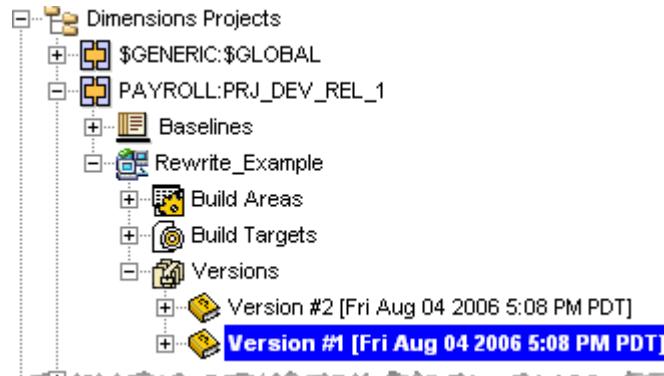
Example of Rewrite

Consider a build configuration with two existing versions:



To Rewrite to a previous version:

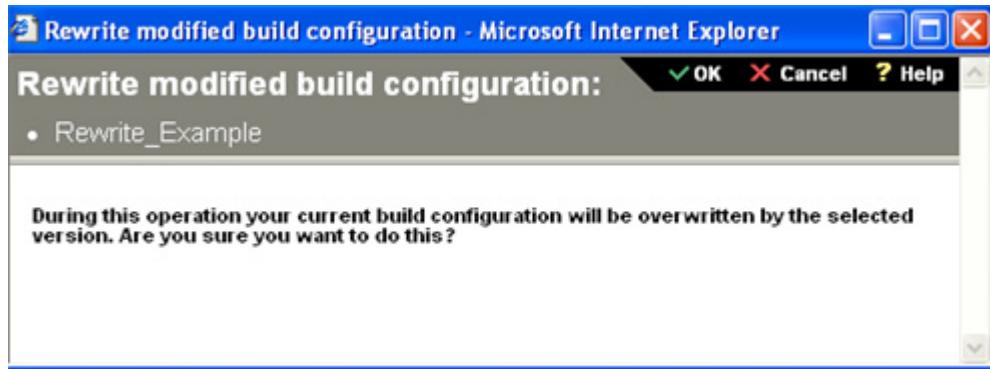
- 1 Select the desired version (in this example, Version #1).



- 2 Click the Rewrite icon.



A warning dialog box appears:



- 3 Click OK to confirm the Rewrite operation. Dimensions replaces the current build configuration with the selected build configuration.

Deleting Versions Not Allowed

You are not allowed to delete a version of a build configuration.

Chapter 12

Scheduling Build Jobs

About Scheduling	250
Managing the Scheduler Service	250
Filtering Dimensions Projects	252
Managing Build Job Schedules	252
Checking the Results of Scheduled Build Jobs	259
Running Builds from the Build Scheduling Tab	259

About Scheduling

This chapter describes how to set up schedules that execute build jobs in the future. You can schedule a build to execute once or to repeat at specific intervals.

When you create a build job schedule you specify the following parameters:

- The build configuration to be executed.
- The build area to be populated.
- The targets to be built.
- When and how often the build job schedule will repeat.

Before you add a build job schedule you must first:

- Create a build configurations, for details see [page 183](#).
- Configure the Dimensions Build Scheduler Service, for details see below.

Managing the Scheduler Service

This section describes how to configure and modify the Dimensions Build Scheduler Service and to verify that the service is running. You must configure the Scheduler Service to enable scheduled build jobs to run.

Configuring the Scheduler Service

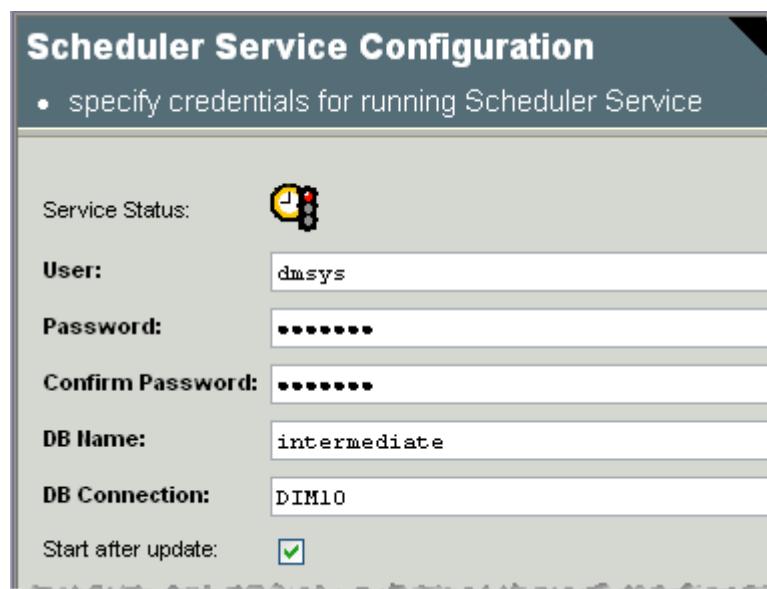
Purpose Follow this procedure to configure the Scheduler Service. You can also use this procedure to modify the current settings of the service.

To configure the Scheduler Service:

- 1 In Dimensions Build click the Build Scheduling tab. In the navigation area click Scheduling.
- 2 In the menu area click Scheduler.



The Scheduler Service Configuration dialog box appears.



- 3 Type the required credentials. Use the same values as the one that you used to log in to the Administration Console. You can also specify the credentials of another user but you must verify that they have sufficient privileges to run a service.
- 4 To restart the Scheduler Service after applying the new or modified credentials, select the **Start after update** check box.
- 5 Click OK.



IMPORTANT!

The Scheduler Service does not restart if Tomcat is restarted

If you restart the Tomcat server the service that controls scheduling does not restart. To automatically restart the scheduling service every time that Tomcat restarts, edit the scheduler.start variable in the following file:

```
<DMRROOT>\Common Tools\tomcat\5.5\webapps\bws\WEB-INF\web.xml
```

Edit the param-value of the scheduler.start variable so that it is set to "1" or "true" and check that this init-param is uncommented.

If you are using a remote build server Dimensions Build may not be able to distinguish between multiple IP addresses. Specify the following variables in the file web.xml:

- The hostname in build.server.host
- The port number in build.server.port

Filtering Dimensions Projects

You can use a filter to limit the Dimensions projects that are displayed in the navigation area. This feature is useful when you have many Dimensions projects and you only want to display a sub-set of them. For details about using filters see [page 235](#).

Managing Build Job Schedules

This section describes how to manage build job schedules.

Viewing Scheduled Build Jobs

Purpose Follow this procedure to view the build jobs that are currently scheduled.

To view scheduled build jobs:

- 1 In Dimensions Build click the Build Scheduling tab.
- 2 To view the build jobs scheduled for a Dimensions project, in the navigation area click Scheduling and click the Dimensions project. The content area lists all the build jobs scheduled for the Dimensions project.
- 3 To view the build jobs scheduled for a build configuration, in the navigation area click the build configuration. The content area lists all the build jobs scheduled for the build configuration.

Adding Build Job Schedules

Purpose Follow this procedure to add a new build job schedule.

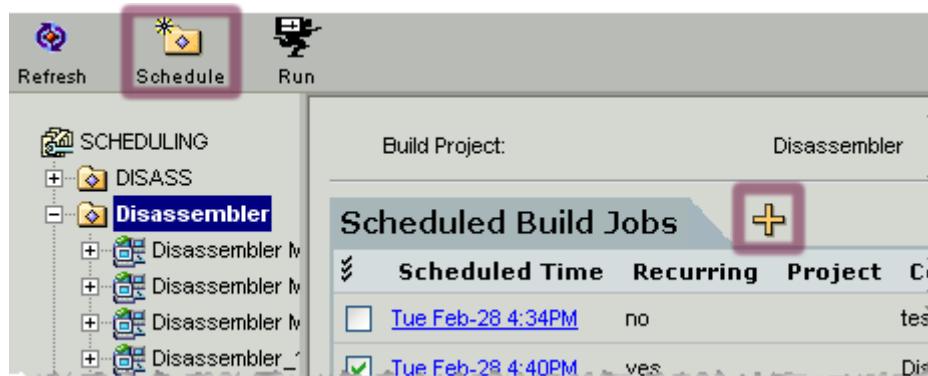


NOTE Before you add a build job schedule you must first create a build configuration and configure the Scheduler Service.

To add a build job schedule:

- 1 In Dimensions Build click the Build Scheduling tab.
- 2 In the navigation area click Scheduling and select the Dimensions project containing the build configuration where you want to add a schedule.

- 3** In the menu area click **Schedule**, or in the Scheduled Build Jobs section of the content area click **New**.



The Add Scheduled Build Job dialog box appears.

- 4** From the **Dimensions Project** list select a Dimensions project.
- 5** From the **Build Configuration** list select a build configuration. To build all configurations select *All configurations*.
- 6** From the **Version** list select the version of the build configuration that you want to build. If you are building all configurations this option is set to *Latest* and you cannot change it.
- 7** From the **Build Area** list select the build area where you want the build configuration to be built. To build all build areas associated with the build configuration select *All areas*. If you are building all configurations this option is set to *All areas* and you cannot change it.

When you setup the build configuration and attached a build area, if you specified that a password is required at runtime, the **Provide passwords for selected areas** link appears to the right of the Build Area list.

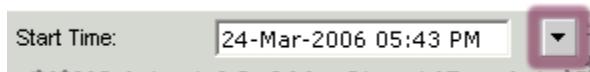


To enable access by the users to the build areas when the scheduled job is run, you must provide a password for each user. Do the following:

- Click **Provide passwords for selected areas**. The Set Build Password(s) dialog box appears.
 - Type and confirm a password for each user.
 - Click Save.
- 8** From the **Build Target** list select the build target to be built. To build all build targets for the build configuration select *All targets*. If you are building all configurations this option is set to *All targets* and you cannot change it.

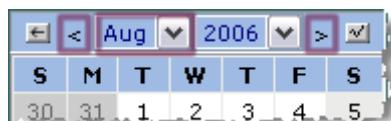
- 9** To specify a start time for the build job schedule do the following:

- a Click the down arrow to the right of the Start Time field.



The calendar picker appears.

- b From the **Month** list select a month, or use the left and right arrows to scroll through the months.

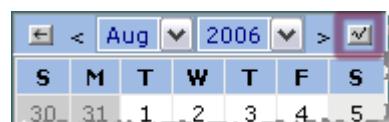


- c From the **Year** list select a year.

- d Select a date. Dates with a light grey background occur on Saturdays and Sundays. Dates with a dark grey background occur in different months and you cannot select them (choose the next or previous month).

- e From the **Time** lists select an hour, minute, and PM or AM.

- f To use this start time click the tick button at the top right corner of the calendar picker.



The start time appears in the Start Time field.

- 10** From the **Time Out** list optionally specify the minutes, hours, or days after which the build job is terminated if it has not finished.
- 11** Select the **Execute clean-up script before build** check box to have the clean-up script defined in the build configuration executed before the build begins (all files in the build area are removed before the build begins).
- 12** Select the **Do not transfer sources** check box if you do not want copies of the source files retrieved to the build area.
- 13** Select the **Do not preserve targets** check box if you do not want the target files to be preserved in Dimensions.
- 14** (If you are preserving targets) From the **Select Dimensions Project for targets preservation** list select the Dimensions project that will receive the target files produced by the build. This list is not displayed if you select the **Do not preserve targets** check box.
- 15** If this is a one-time schedule for a future build job that you do not want to repeat, click OK. The new build job schedule is added to the content area.



NOTE If passwords are required at runtime for the build areas but you have not specified them (see step 7), you will be prompted for the passwords.

To set the schedule to recur see below.

Scheduling the Frequency and Range of Recurring Build Jobs

Purpose	Follow the procedure below to set the frequency and range that a new build job is repeated.
Pre-requisites	Setup a build job schedule, for details see page 252 .

To schedule the frequency of a recurring build job:

- 1 If you are adding a recurrence to a new schedule in the Add Scheduled Build Job dialog box, go to step 2.
If you are adding a recurrence to a new schedule but have closed the Add Scheduled Build Job dialog box, do the following:
 - a In the navigation area click Scheduling and select the Dimensions project containing the build job schedule.
 - b In the Scheduled Build Jobs section of the content area click the scheduled build job. The Edit Scheduled Build Job dialog box appears.
- 2 On the **Scheduled Build Job Details** tab select the **Recurring** check box and click the **Recurrence** tab.
- 3 To specify the frequency that the build job is repeated, select one of the following options:
 - **Hourly**: specify an interval in hours, for example, every 2 hours.
 - **Daily**: specify an interval in days, for example, every 3 days.
 - **Weekly**: specify an interval in weeks and select the day(s) that the build job will repeat. For example, you can specify that the build job will repeat every week on Monday and Friday.
 - **Monthly**: specify the day and the month that the build job will repeat, for example day 16 of month 12.
- 4 Click the **Range of Recurrence** tab.
- 5 To specify when the build job recurrence will start, click the down arrow to the right of the Start field. The calendar picker appears. To select a start time and date do the following:
 - a From the **Month** list select a month, or use the left and right arrows to scroll through the months.
 - b From the **Year** list select a year.
 - c Select a date. Dates with a light grey background occur on Saturdays and Sundays. Dates with a dark grey background occur in different months and you cannot select them (choose the next or previous month).
 - d From the **Time** lists select an hour, minute, and PM or AM.
 - e To use this start time click the tick button at the top right corner of the calendar picker. The start time appears in the Start Time field.

- 6 To specify when the build job recurrence will end, choose one of the following options:
 - **No end date:** the build job recurs indefinitely until you modify or delete the build job schedule.
 - **End after *n* occurrences:** specify the number of times the build job will be repeated.
 - **End by:** select the date that the build job schedule will stop. Click the down arrow to the right of the field and use the calendar picker to select a date (see steps 5a to 5e).
- 7 Click OK.

Modifying Build Job Schedules

Purpose Follow this procedure to modify the parameters of an existing build job schedule.

To modify a build job schedule:

- 1 In Dimensions Build click the Build Scheduling tab.
- 2 In the navigation area click Scheduling and select the Dimensions project containing the build job schedule that you want to modify. In the Scheduled Build Jobs section of the content area, in the Scheduled Time column, click the schedule. The Edit Scheduled Build Job dialog box appears.
- 3 From the **Dimensions Project** list select a Dimensions project.
- 4 From the **Build Configuration** list select a build configuration. To build all configurations select *All configurations*.
- 5 From the **Version** list select the version of the build configuration that you want to build. If you are building all configurations this option is set to *Latest* and you cannot change it.
- 6 From the **Build Area** list select the build area where you want the build configuration to be built. To build all build areas associated with the build configuration select *All areas*. If you are building all configurations this option is set to *All areas* and you cannot change it.

When you attached a build area to the build configuration, if you specified that a password is required at runtime, the **Provide passwords for selected areas** link appears to the right of the Build Area list.



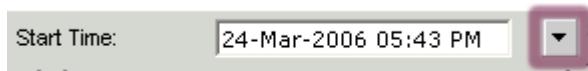
To enable access by the users to the build areas when the scheduled job is run in the future, you must provide a password for each user. Do the following:

- a Click **Provide passwords for selected areas**. The Set Build Password(s) dialog box appears.
- b Type and confirm a password for each user.
- c Click Save.

7 From the **Build Target** list select the build target to be built. To build all build targets for the build configuration select *All targets*. If you are building all configurations this option is set to *All targets* and you cannot change it.

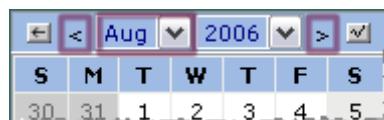
8 To modify the start time for the build job do the following:

- a** Click the down arrow to the right of the Start Time field.

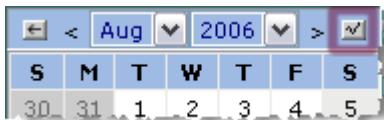


The calendar picker appears.

- b** From the **Month** list select a month, or use the left and right arrows to scroll through the months.



- c** From the **Year** list select a year.
- d** Select a date. Dates with a light grey background occur on Saturdays and Sundays. Dates with a dark grey background occur in different months and you cannot select them (choose the next or previous month).
- e** From the **Time** lists select an hour, minute, and PM or AM.
- f** To use this start time click the tick button at the top right corner of the calendar picker.



The start time appears in the Start Time field.

- 9** From the **Time Out** list optionally specify the minutes, hours, or days after which the build job is terminated if it has not finished.
- 10** Select the **Execute clean-up script before build** check box to have the clean-up script defined in the build configuration executed before the build begins (all files in the build area are removed before the build begins).
- 11** Select the **Do not transfer sources** check box if you do not want copies of the source files retrieved to the build area.
- 12** Select the **Do not preserve targets** check box if you do not want the target files to be preserved in Dimensions.
- 13** (If you are preserving targets) From the **Select Dimensions Project for targets preservation** list select the Dimensions project that will receive the target files produced by the build. This list is not displayed if you select the **Do not preserve targets** check box.
- 14** Click OK.



NOTE If passwords are required at runtime for the build areas but you have not specified them (see step 6), you will be prompted for the passwords.

Modifying the Frequency and Range of Recurring Build Jobs

Purpose Follow the procedure below to modify the frequency and range that a build job is repeated.

To modify the frequency of a recurring build job:

- 1 In Dimensions Build click the Build Scheduling tab.
- 2 In the navigation area click Scheduling and select the Dimensions project containing the build schedule recurrence that you want to modify. In the Scheduled Build Jobs section of the content area click the schedule. The Edit Scheduled Build Job dialog box appears.
- 3 Click the **Recurrence** tab.
- 4 To modify the frequency that the build job is repeated select one of the following options:
 - **Hourly**: specify an interval in hours, for example, every 2 hours.
 - **Daily**: specify an interval in days, for example, every 3 days.
 - **Weekly**: specify an interval in weeks and select the day(s) that the build job will repeat. For example, you can specify that the build job schedule will repeat every week on Monday and Friday.
 - **Monthly**: specify the day and the month that the build job will repeat, for example day 16 of month 12.
- 5 Click the **Range of Recurrence** tab.
- 6 To modify the start time of the build job recurrence, click the down arrow to the right of the Start field. The calendar picker appears. To select a start time and date do the following:
 - a From the **Month** list select a month, or use the left and right arrows to scroll through the months.
 - b From the **Year** list select a year.
 - c Select a date. Dates with a light grey background occur on Saturdays and Sundays. Dates with a dark grey background occur in different months and you cannot select them (choose the next or previous month).
 - d From the **Time** lists select an hour, minute, and PM or AM.
 - e To use this start time click the tick button at the top right corner of the calendar picker. The start time appears in the Start Time field.
- 7 To modify when the build job recurrence will end, choose one of the following options:
 - **No end date**: the build job recurs indefinitely until you modify or delete the build job schedule.
 - **End after n occurrences**: specify the number of times the build job will be repeated.
 - **End by**: select the date that the build job will stop. Click the down arrow to the right of the field and use the calendar picker to select a date (see steps 6a to 6e).
- 8 Click OK.

Deleting Build Job Schedules

Purpose Follow this procedure to delete build job schedules.

To modify build job schedules:

- 1 In Dimensions Build click the Build Scheduling tab.
- 2 In the navigation area click Scheduling and click the Dimensions project containing the build job schedules that you want to delete.
- 3 In the content area select the build job schedules and click Delete.

	ID	Scheduled Time	Recurring	Project	Configuration
<input checked="" type="checkbox"/>	436	Thu May-11 5:10PM	no	PAYROLL:DISASSEMBLER	Disassembler
<input type="checkbox"/>	435	Wed May-10 5:09PM	yes	PAYROLL:DISASSEMBLER	Disassembler

- 4 Click Yes to confirm that you want to delete the build job schedules.

Checking the Results of Scheduled Build Jobs

To check the progress and results of scheduled build jobs use the Build Monitoring tab. For details see [Chapter 13, "Monitoring Builds in the Build Administrator" on page 261](#).

Running Builds from the Build Scheduling Tab

You normally run builds from the Build Management tab. However you can also run builds from the Build Scheduling tab.



NOTE To run a build configuration you must first check it in.

To run a build from the Build Scheduling tab:

- 1 In the navigation area of the Build Scheduling tab click Scheduling and select the Dimensions project containing the build configuration that you want to run.
- 2 On the menu area click Run. The Build wizard appears. For details about using the wizard see [page 228](#).
- 3 Click Run. You can monitor the progress of the build in the Build Monitoring tab.

Chapter 13

Monitoring Builds in the Build Administrator

About Monitoring Builds	262
Reviewing Build Status Immediately After Launch	262
Viewing the Build Job Details	265
Monitoring a Running Build	268
Canceling a Running Build Job	269
Rebuilding the Selected Build Job	272
Viewing Build Execution History	270
Viewing Build Launch Attempts	271
Viewing Dependencies	273
Viewing a Bill of Materials	275

About Monitoring Builds

This section describes how to use the Build Monitoring features of Dimensions. These features allow you to examine details such as:

- what time the build started
- what time the build completed
- whether or not there was an error
- what Dimensions project and build configuration were used
- the dependencies of the build job
- Bill of Materials information
- who initiated the build

In addition to these features, Dimensions also allows you to rebuild previous build jobs, and to cancel running build jobs. These features are discussed after the basic build monitoring features.

In general, build monitoring falls into three categories:

- Reviewing build status immediately after launch
- Monitoring the status of a running build
- Reviewing the history of previous builds

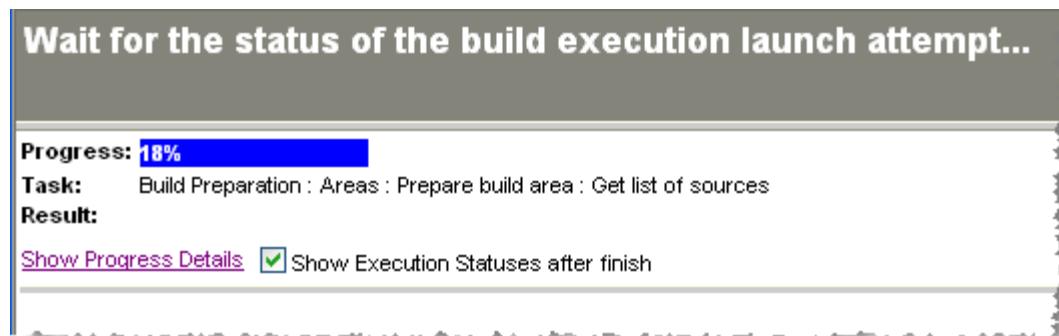
To review build status after you launch a build, use the two monitoring icons that appear in the **Build Execution Status** dialog box.

To monitor a running build or to review the history of previous builds, use the **Build Monitoring** tab of the Administration Console.

A discussion of these three monitoring methods follows.

Reviewing Build Status Immediately After Launch

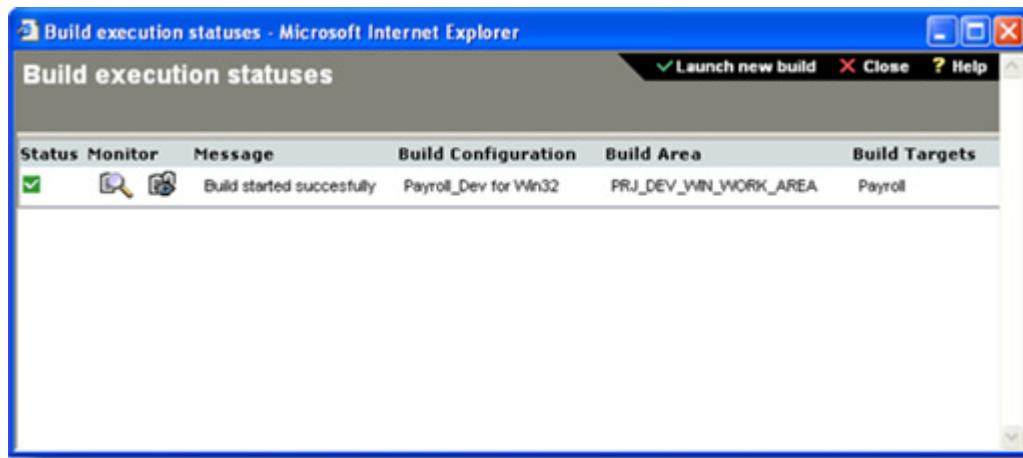
After you launch a build, the **Wait for the status of the build execution launch attempt** dialog box appears. A progress bar shows the percentage of launch activity that is complete. When the progress bar reaches 100%, the build is ready to begin.



If you click Show Progress Details, you can view the high-level details of the launch preparation, such as the following:

```
Build Preparation
Build Preparation : Areas
Build Preparation : Areas : Prepare build area
Build Preparation : Areas : Prepare build area : Get list of sources
Completed
Build Preparation : Areas : Prepare build area : Transfer sources
transfer completed
area completed HELLOWORLD_AREA
Build Preparation
Build Preparation completed
```

If you check the box for **Show Execution Statuses after finish**, the **Build Execution Status** dialog box appears.



Notice that in this dialog box, two icons appear under the Monitor heading:



- The left icon, with the magnifying glass, displays the launch attempt log.
- The right icon, with the eye, displays the build job details.

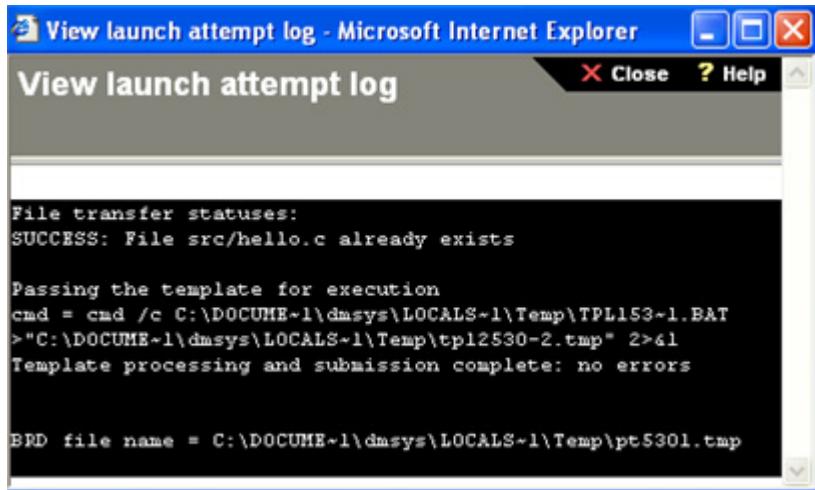
If you do not select the **Show Execution Statuses after finish** check box, you can still cause the dialog box to appear by clicking the **Show Execution Status** link on the **Wait for the status of the build execution launch attempt** dialog box.



NOTE On mainframe machines, build jobs do not always execute immediately and may be queued. Dimensions will wait a certain period of time for a response from the remote node executing the job. If the job execution occurs after this amount of time, Dimensions is unable to display any information in the Build Execution Statuses dialog box.

Viewing the Launch Attempt Log

If you click the left icon under **Monitor** on the **Build Execution Status** dialog box, the **View launch attempt log** dialog box appears:



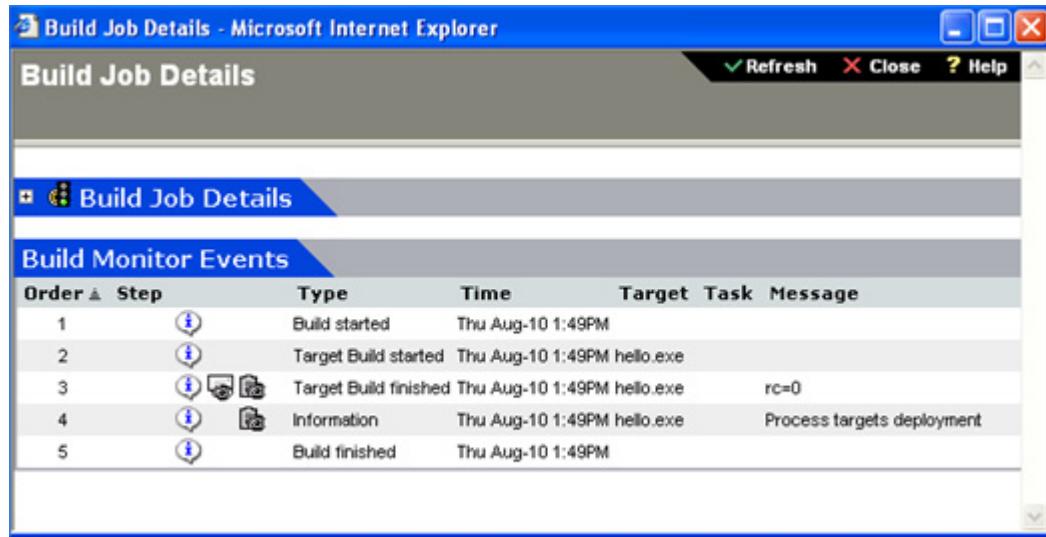
The launch attempt log shows details of the attempt by Dimensions launch the build job. If a part of the build failed, you will likely see an indication of it here. In the example shown, you can see the following:

- The one source file for the build (hello.c) was found to already exist and thus was not transferred to the build area.
- The script or template for building the target was processed into a temporary batch file and was found to have no errors.
- The BRD file, which contains information such as the values of build options, was generated and stored in a temporary directory. (See [Preserving Temporary Files](#) on page 317 for details on how to use this file.)

As a general rule, the information in the launch attempt log is more suitable for debugging a build that is not launching properly. To see details of a build that launches, but completes with errors, consult the Build Job Details.

Viewing the Build Job Details

If you click the See Build Events icon, in the **Monitor column**, on the **Build Execution Status** dialog box, the **Build Job Details** dialog box appears:



By default, the Build Job Details section is hidden and the Build Monitor Events section is displayed.

Interpreting Build Monitor Events

The Build Monitor Events section displays a table of events that occurred during the build. Each row represents an event. Earlier events are displayed at the top of the table; later events at the bottom.

Build events typically show events such as when the build started, when the build of specific targets started and finished, and errors that occurred during specific steps.

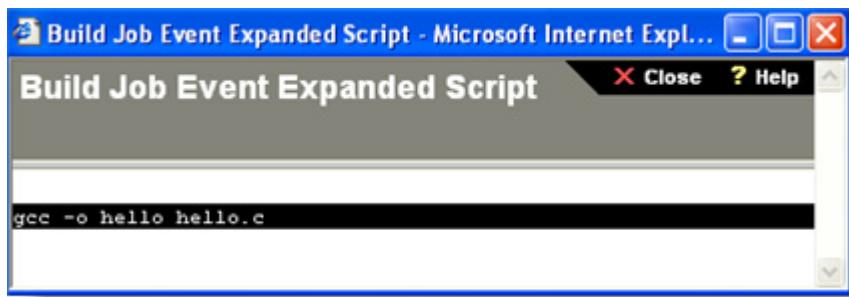
Build Monitor Events						
Order	Step	Type	Time	Target	Task	Message
1		Build started	Thu Aug-10 2:30PM			
2		Target Build started	Thu Aug-10 2:30PM	hello.exe		
3		Target Build finished	Thu Aug-10 2:30PM	hello.exe		rc=1
4		Build finished	Thu Aug-10 2:30PM			

Specific events may display one of the following icons:

Icon	Description
	Display the Build Job Event Expanded Script dialog box. This displays a command-prompt-type display with the script for the build step in question.
	Display the Build Job Event Output Log dialog box. This log displays information such as file deployment statuses, the success of the step, and so on.
	Display the Build Job Event Error Log dialog box. This log displays error information for build steps that fail.

Expanded Script

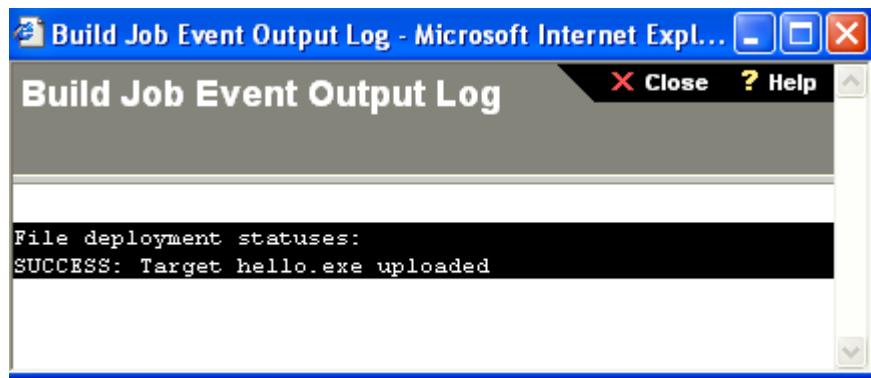
Here is an example of the expanded script:



The expanded script displays the build execution command found in the build scripts.

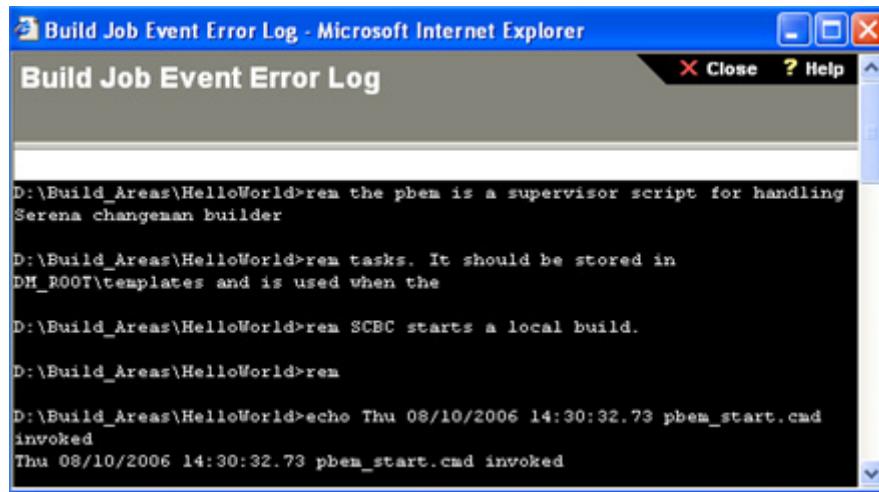
Build Job Event Output Log

Here is an example of the build job event output log.



Error Log

Here is an example of the error log:



```
D:\Build_Areas\HelloWorld>rem the pbem is a supervisor script for handling
Serena changeman builder

D:\Build_Areas\HelloWorld>rem tasks. It should be stored in
DM_ROOT\templates and is used when the

D:\Build_Areas\HelloWorld>rem SCBC starts a local build.

D:\Build_Areas\HelloWorld>rem

D:\Build_Areas\HelloWorld>echo Thu 08/10/2006 14:30:32.73 pbem_start.cmd
invoked
Thu 08/10/2006 14:30:32.73 pbem_start.cmd invoked
```

In the error log you can find information on which tasks failed:

```
16PB0011I Task 1 executed with errc = 1 - failed
16PB0044I Processing rc: 1; Highest Significant Subtask Rc: 1
16PB0045I Of the 1 tasks requested; 0 succeeded, 1 failed, and 0 were
skipped
```

Viewing Build Details

If you click the plus sign next to the Build Job Details heading, the build details section expands to show its full information:

Build Job Details	
Build Job ID:	295
Started at:	Thu Aug-10 1:49PM
Finished at:	Thu Aug-10 1:49PM
Initiated by:	dmsys
Build Result:	Build job executed successfully
Dimensions Project:	PAYROLL:PRJ_INITIAL
Build Configuration:	HelloWorld
Platform:	Win32
Targets:	Hello Executable;

This section contains information such as who initiated the build, which Dimensions project and build configuration were used, and what platform the build was for.

Monitoring a Running Build

Other sections have described how to look at the details of a build you just launched. If the build is a long-running job, you can also examine its details using the Build Monitoring tab.

To monitor a running build:

- 1** Log on to the Dimensions administration console.
- 2** Click the Build Monitoring tab. In the left part of the Administration Console, you see the tree with Running Jobs and History entries.
- 3** Click Running Jobs. The right side of the Administration Console now displays a list of the active (currently running) build jobs:

Status	Time Start	Build Configuration	Version
	Thu Aug-10 2:46PM	Monitoring_Test	1



NOTE If instead of clicking Running Jobs you click the plus sign to the left of Running Jobs, the list of build jobs appears in the object tree only.

Notice also that the top of the display now shows three icons—Refresh, Cancel, and Filter:

- **Refresh** refreshes the display.
- **Cancel** cancels the selected build jobs. You can cancel more than one build job at a time.
- **Filter** applies criteria that you define to limit what is displayed. Defining a filter is covered in [Using Filters](#) on page 235.

- 4 Click the build job that you are interested in. The Build Job Details and Build Monitor Events appear on the right, in the content area.

The screenshot shows the Dimensions Administration Console interface. On the left, there's a navigation pane with 'Running Jobs' and 'History'. The main content area has two tabs: 'Build Job Details' and 'Build Monitor Events'. The 'Build Job Details' tab is active, displaying information about a build job with ID 297, started on Thu Aug-10 2:46PM by dimsyst. The 'Build Monitor Events' tab shows a table of events:

Order	Step	Type	Time	Target
1		Build started	Thu Aug-10 2:46PM	
2		Target Build started	Thu Aug-10 2:46PM	TargetOne

Cancelling a Running Build Job

You can cancel a running build job from the Build Monitoring tab of the Dimensions Administration Console.

To cancel a running build job:

- 1 In the Dimensions Administration Console, click the Build Monitoring tab.
- 2 Click Running Jobs to display the list of running jobs.
- 3 Select the check box to the left of the build job you want to delete. The Cancel icon turns red.



- 4 Click the Cancel icon. Dimensions displays a confirmation dialog box and then deletes the selected build job.

Monitoring Past Builds

When you want to view the history of build jobs that have been launched in the past, you can use the features available under the Build Monitoring tab.

Viewing Build Execution History

To view the execution history for a past build:

- 1 In the Dimensions Administration Console, click the Build Monitoring tab.
- 2 Click History to display a list of Dimensions projects.
- 3 Click the blue link that corresponds to the Dimensions project whose execution history you want to view. The Execution History tab and Launch Attempts tab appear on the right side of the application.

ID	Status	Time Start	Time Stop	Build Configuration
298		Thu Aug-10 2:51PM	Thu Aug-10 2:53PM	Monitoring_Test
297		Thu Aug-10 2:48PM	Thu Aug-10 2:48PM	Monitoring_Test
296		Thu Aug-10 2:30PM	Thu Aug-10 2:30PM	HelloWorld
295		Thu Aug-10 1:49PM	Thu Aug-10 1:49PM	HelloWorld
286		Thu Aug-10 11:20AM	Thu Aug-10 11:20AM	HelloWorld
285		Thu Aug-10 11:18AM	Thu Aug-10 11:18AM	HelloWorld

The status column shows whether or not the build completed successfully.

- 4 To see more details of a particular build execution:
 - a Click Execution History if it is not selected already.
 - b Click the link corresponding to the instance in question.

ID	Status	Time Start
295		Thu Aug-10 1:49PM
298		Thu Aug-10 2:51PM

The Build Monitor Events for that build execution display.

The screenshot shows two windows side-by-side. The left window is titled 'Build Job Details' and contains the following information:

Build Job ID:	295
Started at:	Thu Aug-10 1:49PM
Finished at:	Thu Aug-10 1:49PM
Initiated by:	dmsys
Build Result:	Build job executed successfully
Dimensions Project:	PAYROLL:PRJ_INITIAL
Build Configuration:	HelloWorld
Platform:	Win32
Targets:	Hello Executable;

The right window is titled 'Build Monitor Events' and displays a table of events:

Order	Step	Type	Time	Target
1		Build started	Thu Aug-10 1:49PM	
2		Target Build started	Thu Aug-10 1:49PM	hello.exe
3		Target Build finished	Thu Aug-10 1:49PM	hello.exe
4		Information	Thu Aug-10 1:49PM	hello.exe
5		Build finished	Thu Aug-10 1:49PM	

Viewing Build Launch Attempts

You can view a list of the launch attempts for a build job.

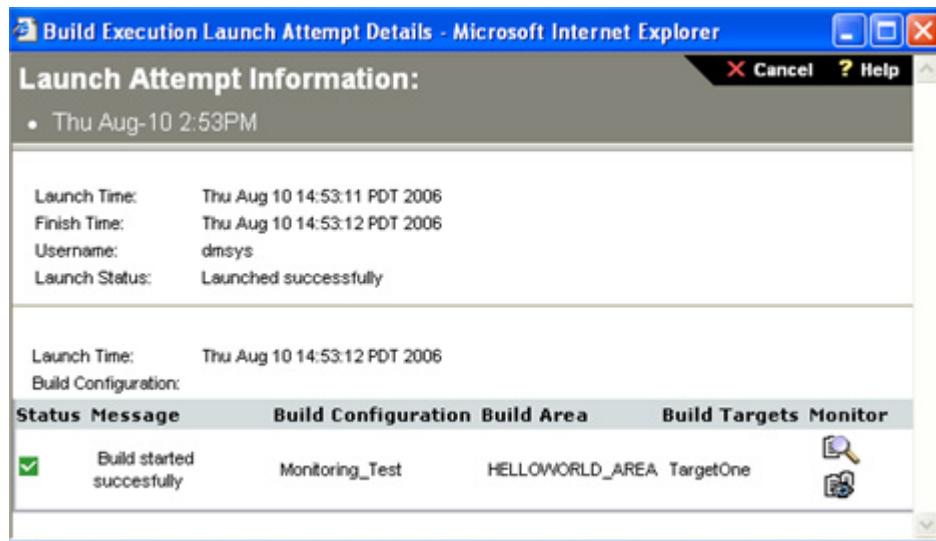
To view build launch attempts:

- 1 Follow the procedure in [Viewing Build Execution History](#) on page 270, through Step 3.
- 2 Click the Launch Attempts tab. The list of launch attempts appears.

The screenshot shows a table titled 'Launch Attempts' with the following data:

	Status	Launch Time	Finish Time	Originator
<input type="checkbox"/>	568	<input checked="" type="checkbox"/> Thu Aug-10 2:53PM	Thu Aug-10 2:53PM	dmsys
<input type="checkbox"/>	567	<input checked="" type="checkbox"/> Thu Aug-10 2:52PM	Thu Aug-10 2:52PM	dmsys
<input type="checkbox"/>	566	<input checked="" type="checkbox"/> Thu Aug-10 2:51PM	Thu Aug-10 2:51PM	dmsys
<input type="checkbox"/>	565	<input checked="" type="checkbox"/> Thu Aug-10 2:46PM	Thu Aug-10 2:46PM	dmsys
<input type="checkbox"/>	564	<input checked="" type="checkbox"/> Thu Aug-10 2:30PM	Thu Aug-10 2:30PM	dmsys
<input type="checkbox"/>	563	<input checked="" type="checkbox"/> Thu Aug-10 1:49PM	Thu Aug-10 1:49PM	dmsys
<input type="checkbox"/>	544	<input checked="" type="checkbox"/> Thu Aug-10 11:20AM	Thu Aug-10 11:20AM	dmsys
<input type="checkbox"/>	543	<input checked="" type="checkbox"/> Thu Aug-10 11:18AM	Thu Aug-10 11:18AM	dmsys

- 3 Click the blue link for a given launch attempt to display a dialog box with further details:



Rebuilding the Selected Build Job

You can rebuild a build job whose history you are viewing from the Build Monitoring tab.

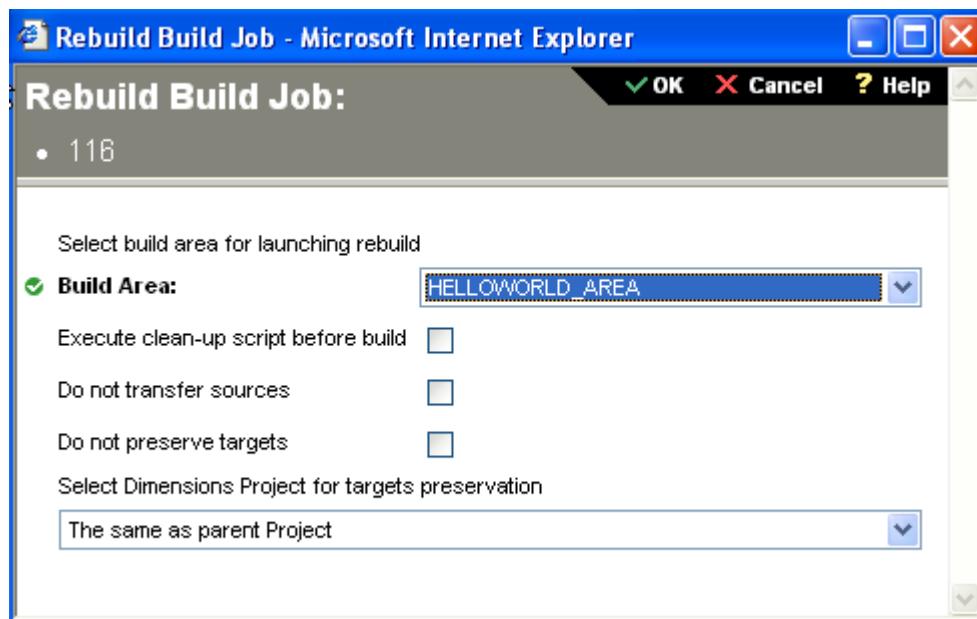
To rebuild the selected build job:

- 1 In the Dimensions Administration Console, click the Build Monitoring tab.
- 2 Click Running Jobs to display the list of running jobs, or History to display previously-executed jobs.
- 3 Be sure Execution History is selected, not Launch Attempts.
 - a If you are viewing jobs from the Running Jobs list, select the build job you want to rebuild.
 - b If you are viewing jobs from the History list, select the check box next to the build job you want to rebuild.

In either case, the Rebuild icon appears:



- 4 Click the Rebuild icon. The Rebuild Build Job dialog box appears:



- 5 Select a different build area from the drop-down list, or click OK to accept the default.
- 6 Check the box for **Execute clean-up script before build** if you want to execute the script named or defined in the Clean-up script area of the build configuration. (If not such script is defined, nothing will happen.)
- 7 Check the box for **Do not transfer sources** and **Do not preserve targets** as desired.
- 8 If you preserve targets, select the Dimensions project that should receive the targets, or accept the default of **The same as parent Project**.
- 9 Click OK. The **Wait for statuses of build execution launch attempt** dialog box appears, followed by the **Build execution statuses** dialog box.

Viewing Dependencies

You can view build dependencies from the Build Monitoring tab. The dependencies show which modules or items were used to create the final targets.



NOTE To display dependencies, you must configure the build job to preserve targets.

To view the dependencies for a specific target:

- 1 Navigate to the Build Monitoring tab.
- 2 Expand the History tree until you find the versions, and underneath them, the specific build execution (a date and time) you are interested in.

- 3 Expand the build execution record and the dependencies and bill of materials entries appear:



NOTE Only successful build executions will display dependencies. If the traffic light symbol is not green, no dependencies will be visible.

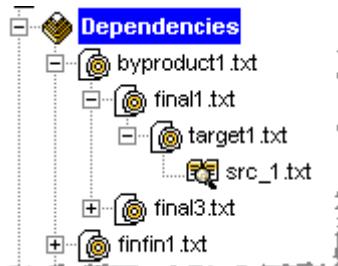
- 4 Select the Dependencies entry. The Job Targets and Job Sources listings appear in the content area.

Job Targets			
Name	Relative Path	Dimensions	Location
byproduct1.txt			QLARIUS:BYPRODUCT1 TXT~01.A-SRC;3
final1.txt			QLARIUS:FINAL1 TXT~01.A-SRC;3
final3.txt			QLARIUS:FINAL3 TXT~01.A-SRC;3
finfin1.txt			QLARIUS:FINFIN1 TXT~01.A-SRC;3
finfin2.txt			QLARIUS:FINFIN2 TXT~01.A-SRC;3
policytest1.virt			QLARIUS:POLICYTEST1 VIRT~02.A-DAT;1.6
policytest3.place			QLARIUS:POLICYTEST3 PLACE~02.A-DAT;1.6
sidefile.flum			QLARIUS:SIDEFILE FLUM~02.A-DAT;1.6
target1.txt			QLARIUS:TARGET1 TXT~02.A-SRC;7
target2.txt			QLARIUS:TARGET2 TXT~02.A-SRC;3
target3.txt			QLARIUS:TARGET3 TXT~02.A-SRC;3

Job Sources			
File	Relative Path	Dimensions	Location
src_1.txt			QLARIUS:SRC 1 TXT.A-SRC;1
src_2.txt			QLARIUS:SRC 2 TXT.A-SRC;1
src_3.txt			QLARIUS:SRC 3 TXT.A-SRC;1

These listings indicate the targets and sources upon which the build job depends.

- 5 To view the hierarchy of dependencies for a specific file, expand the entries in the Dependencies tree:



Selecting specific entries in hierarchies such as the one shown causes the content area to display source/target details, and dependent sources and targets.

Viewing a Bill of Materials

You can view a bill of materials for some Dimensions projects. The bill of materials contains the list of the items or modules that are included in a final target.

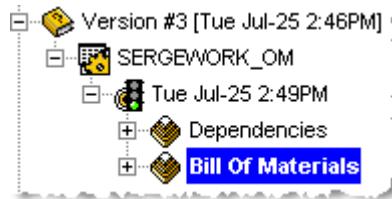
Bill of Materials are available for builds created from an Openmake build, or from a mainframe build.

To configure bill of materials information:

- Bills of materials for mainframe builds can be configured using the SBEM templates. See [Dimensions Build Utility Programs](#) on page 261, and the *Serena Dimensions CM Developer's Reference*.
- Bills of materials for Openmake builds must have the variable DMOBOMRPT set in a template. This variable should point to the location of the bill of materials generated by Openmake. See [Using Dimensions Build with Openmake](#) on page 301.

To view the bill of materials:

- 1 Navigate to the Build Monitoring tab.
- 2 Expand the History tree until you find the versions, and underneath them, the specific build execution (a date and time) you are interested in.
- 3 Expand the build execution record and the dependencies and bill of materials entries appear:



NOTE Only successful build executions will display bill of materials information. If the traffic light symbol is not green, no bill of materials information will be visible.

- 4 Select the Bill of Materials entry. The Job Details and Job Summaries listings appear in the content area:

Bill of Materials: Job Details

Started at:	Tue Jul 25 14:49:37 PDT 2006
Finished at:	Tue Jul 25 14:49:58 PDT 2006
Initiated by:	dmsys
Build Result:	0
Dimensions Project:	TEST:SERGE
Build Configuration:	sergeDemo_om
Platform:	Win32
Targets:	jar files;

Bill of Materials Job Summaries

Start Time	Project Name	Search Path
Tue Jul-25 2:50PM	COM.SERENA.BUILD.CORE	DEV

- 5 Click a job summary link to view the Details and Targets information:

Bill of Materials: Details

Start Time:	Tue Jul-25 2:50PM
Project Name:	COM.SERENA.BUILD.CORE
Search Path:	DEV
Control File Name:	D:/TestDev/WorkArea/openmake/work\2006072521493
Log URL:	http://pc2057:58080/openmake/OMDisplayLog?BuildJob=DEV&Machine=pc2057&Date Time=2006-07-25 14:49:37
OS:	windows

Bill of Materials: Targets

Name	Path
com.serena.build.core.classpath	.
com.serena.build.core.jar	.
com.serena.build.core.javac	.

Chapter 14

Managing Notifications

About Notifications	278
Managing the Notification Service	280
Managing Notification Templates	281
Managing Notification Subscriptions	285

About Notifications

Dimensions Build includes a notification facility that enables you to send e-mail messages containing information about build events. For example, you can notify users when a build is completed or when a build area is updated.

Formatting Notification Templates and Events

The format of the e-mail messages generated by notification events is in HTML. To customize the look of these messages you can insert any HTML formatting tag, such as `
` and ``, into the message field of your templates and events. You do not have to specify the opening and closing `<html>` and `<body>` tags.



TIP To ensure that e-mail message can be read easily, add the line break tag `
` at the end of each line in your message, for example:

```
Build configuration name <%CONFIG%><br>
Build configuration prescript <%CONFIG_PRESCRIPT%><br>
Build configuration main script <%CONFIG_MAINSCRIPT%><br>
```

Notification Templates

You can create notification templates and use them when you add notification events. Notification templates can include freeform text and any of the pre-defined variables. Templates are useful when you want to re-use the same subject and message in multiple notification events.

In the example below the template includes the following variables:

- `<%PROJECT%>`: displays the name of the parent Dimensions project.
- `<%CONFIG%>`: displays the name of the build configuration where the notification event is setup.
- `<%JOB_AREA%>`: displays the name of the build area.
- `<%JOB_START_TIME%>`: displays the time that the build job started.
- `<%JOB_STOP_TIME%>`: displays the time that the build job finished.

Template:	Build Finished	
Subject:	Build result for <%PROJECT%>:<%CONFIG%>	<input type="button" value="Insert variable"/>
Message:	<pre>The build has finished successfully for the following build configuration:
Project: <%PROJECT%>
Build configuration: <%CONFIG%>
Build Area: <%JOB_AREA%>
Start time: <%JOB_START_TIME%>
Finish time: <%JOB_STOP_TIME%></pre>	

Notification Events

A notification event specifies the action that causes an e-mail message to be sent to the users that are subscribed to it. You can optionally use notification templates to automate and simplify the process of setting up events.

For each notification event you specify the following:

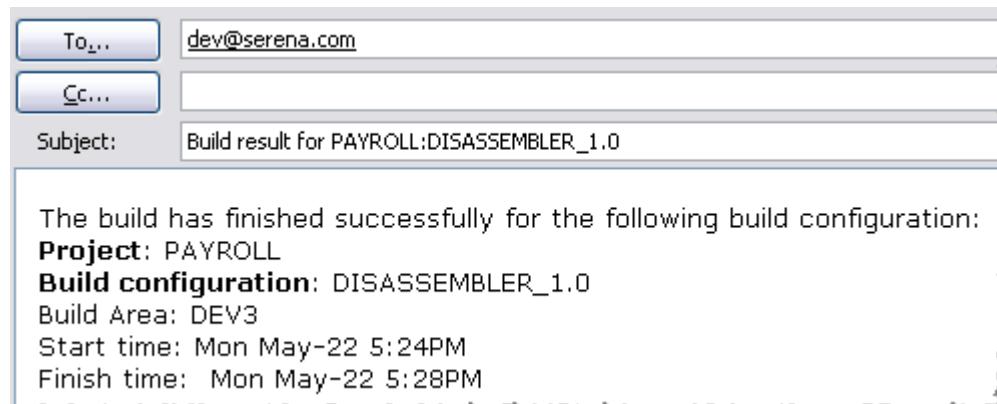
- The Dimensions project, build configuration, and targets to which the event applies. You can specify an individual target or all targets.
- The type of event that triggers the notification. You can choose any of the pre-defined event types, for example:
 - When a build is started.
 - When the targets are built.
 - When there are errors.
- (Optional) The notification template to be used in the e-mail message.
- The subject of the e-mail message. Not required if you are using a template that includes a subject.
- The body of the e-mail message. Not required if you are using a template that includes the message.

Notification Example

In the example below, the notification is sent to the event subscribers after the build of all the targets in *Disassembler_1.0* finishes successfully. The template used is *Build Finished*. No subject and body are required as they are taken from the template:

Project:	Disassembler	
Build Configuration:	Disassembler_1.0	
Target:	All targets	
Event:	Build is finished	
Build Result:	SUCCESS	
Template:	Build Finished	
Subject:	<input type="text"/>	
Message:	<input type="text"/>	
Insert variable		

The e-mail message for this example, using the template *Build Finished* described on page 278, looks like this:



Notification Subscriptions

After you have created a notification event you add subscribers to it. Subscribers can be individual Dimensions users or groups of users. After you add the subscribers they will start to receive e-mail messages based on the notification event criteria that you specified.

Managing the Notification Service

This section describes how to configure and modify the Dimensions Build Notification Service. You must configure the Notification Service to be able to send e-mail notifications.

Configuring the Notification Service

Purpose Follow this procedure to create or modify the parameters required to use an SMTP server to send e-mail notifications.

To configure the Notification Service:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications. In the menu area click Notification. The Notification Service Configuration dialog box appears.
- 2 For **Host** type the name of the node hosting the e-mail server that will send the notifications.
- 3 For **Port** type the port number that the e-mail server is listening on.
- 4 For **Hello host** type the string required to establish a connection through the SMTP protocol.
- 5 For **User** type the name of a user that can access the e-mail server.

- 6** For **Password** type the password for the user that you specified in the previous step.
For **Confirm Password** retype the password.
- 7** From the **Authentication** list select one of the following SMTP server authentication methods:
 - <none>
 - LOGIN
 - PLAIN
 - CRAM-MD5



NOTE If you select <none> you do not need to specify a user and password.

- 8** For **Sender Mail** type the e-mail address from who the notification e-mails will be sent.
- 9** For **Sender Name** type the name that will appear in the From field in the notification e-mails.
- 10** Click OK.

Managing Notification Templates

Viewing Notification Templates

Purpose Follow this procedure to view a list of all the notification templates that are currently defined.

To view notification templates:

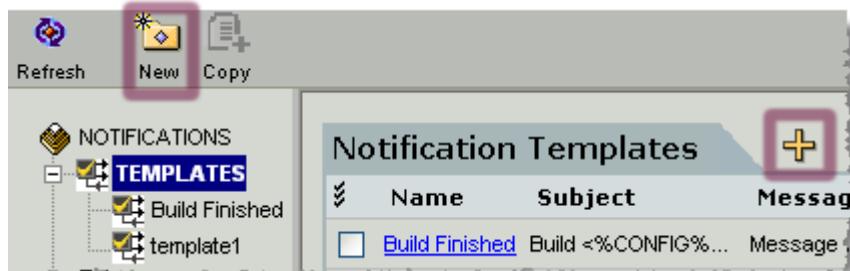
- 1** In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Templates. The navigation and content areas refresh and display all the notification templates that are currently defined.
- 2** To view the details of a template, select it in the navigation and content areas. The navigation and content areas refresh and display details of the notification template.

Adding Notification Templates

Purpose Follow this procedure to add a new notification template.

To add a notification template:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Templates. In the menu or content areas click New Object.



The Add New Notification Template dialog box appears.

- 2 For **Template** type a name for this notification template.
- 3 For **Subject** type the subject of this notification template. To add a pre-defined variable do the following:
 - a Place the cursor where you want to add the variable.
 - b Click **Insert Variable**. The Insert Notification Variable to the Field dialog box appears.
 - c From the **Variable Name** field select a variable. The content of the variable is displayed in the Preview field. The **Description** field briefly describes the variable.
 - d Click OK.



TIP After you have inserted variables you can add freeform text anywhere in the field. You can also cut and paste variables if you want to rearrange their order.

- 4 For **Message** type the message for this notification template. Add a pre-defined variable if required. Add HTML formatting tags if required.
- 5 Click OK. The new notification template is added to the content area.

Editing Notification Templates

Purpose Follow this procedure to edit an existing notification template.

To edit a notification template:

- 1** In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Templates. In the menu or content areas click the name of the notification template that you want to edit.
- 2** In the content area click Edit Notification Template Information.



The Edit Notification Template dialog box appears.

- 3** For **Template** modify the name of this notification template.
- 4** For **Subject** modify the subject of this notification template. To add a pre-defined variable do the following:
 - a** Place the cursor where you want to add the variable.
 - b** Click **Insert Variable**. The Insert Notification Variable to the Field dialog box appears.
 - c** From the **Variable Name** field select a variable. The content of the variable is displayed in the Preview field. The **Description** field briefly describes the variable.
 - d** Click OK.



TIP After you have inserted variables you can add freeform text anywhere in the field. You can also cut and paste variables if you want to rearrange their order.

- 5** For **Message** modify the message for this notification template. Add pre-defined variables and HTML formatting tags if required.
- 6** Click OK. The details of the notification template are updated in the content area.

Copying Notification Templates

Purpose Follow this procedure to copy an existing notification template.

To copy a notification template:

- In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Templates. In the content area select the check box next to the notification template that you want to copy. Click Copy.



The Create New Copy of the Notification Template dialog box appears.

- For **Template** type a name for the new notification template.
- For **Subject** modify the subject of this notification template. To add a pre-defined variable do the following:
 - Place the cursor where you want to add the variable.
 - Click **Insert Variable**. The Insert Notification Variable to the Field dialog box appears.
 - From the **Variable Name** field select a variable. The content of the variable is displayed in the Preview field. The **Description** field briefly describes the variable.
 - Click OK.



TIP After you have inserted variables you can add freeform text anywhere in the field. You can also cut and paste variables if you want to rearrange their order.

- For **Message** modify the message for this notification template. Add pre-defined variables and HTML formatting tags if required.
- Click OK. The new notification template is added to the content area.

Deleting Notification Templates

Purpose Follow this procedure to delete notification templates.

To delete notification templates:

- In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Templates.
- In the content areas select the check box next to each notification template that you want to delete.
- Click Delete. To confirm that you want to delete the notification templates click Yes.

Managing Notification Subscriptions

Filtering Dimensions Projects

You can use a filter to limit the Dimensions projects that are displayed in the navigation area. This feature is useful when you have many Dimensions projects and you only want to display a sub-set of them. For details about using filters see [page 235](#).

Viewing Notification Events, Details, and Subscribers

Purpose Follow this procedure to view a list of all the notification events that are currently defined for a Dimensions project. You can also view details of individual notification events including lists of subscribers.

To view notification events, details, and subscribers:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Subscriptions.
- 2 In the navigation or content areas click the Dimensions project containing the notification events. The content area refreshes and displays all the notification events that are currently defined for the build configurations in that Dimensions project. For example:

Notification Events		
Event Type	Project	Build Config
<input type="checkbox"/> Build is finished	Disassembler	Disassembler Mainframe Build
<input type="checkbox"/> Trace	Disassembler	Disassembler Mainframe Build v2.0
<input type="checkbox"/> Information	Disassembler	Disassembler Mainframe Build
<input type="checkbox"/> Error	Disassembler	Disassembler 1.0

- 3 To view the notification events defined for a specific build configuration, in the navigation area expand the Dimensions project and click the build configuration. The content area refreshes and displays the notification events that are currently defined for that build configuration.
- 4 To view the details of an individual notification event, in the navigation area expand the parent build configuration, expand Events, and click the notification event. The content area refreshes and displays the following information:
 - The details of the notification event.
 - The users and groups that are subscribed to the notification event.

Adding Notification Events

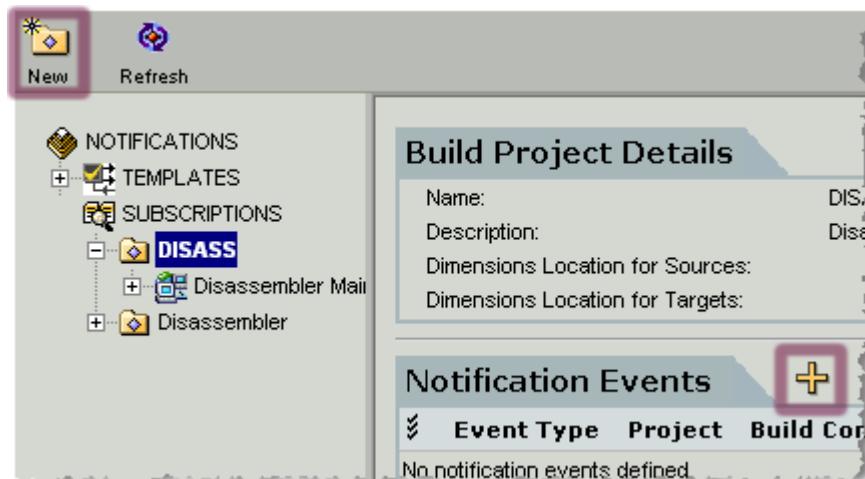
Purpose Follow this procedure to add a new notification event.



NOTE If you do not specify a subject, message, or notification template the notification event is ignored.

To add a notification event:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Subscriptions. In the navigation or content areas click the Dimensions project containing the build configuration where you want to add a notification event. In the menu or content areas click Create New Notification Event.



The Add New Notification Event dialog box appears.

- 2 From the **Project** list select the Dimensions project where you want to add a notification event.
- 3 From the **Build Configuration** list select the build configuration where you want to add a notification event.
- 4 From the **Targets** list select the target to which the notification event will apply. To trigger a notification for all targets select *All Targets*.
- 5 From the **Event** list select the type of event that will trigger the notification event. Some events also require that you select an additional option. For example, if you select the *Build is finished* event type you must also select an option from the Build Result list.
- 6 From the **Template** list optionally select a notification template that will be used to populate the subject and message fields in the e-mail message. If you do not want to use a template select *<none>*.
- 7 For **Subject** type the subject of this notification event. If you do not specify a subject the template subject is used. To add a pre-defined variable do the following:
 - a Place the cursor where you want to add the variable.
 - b Click **Insert Variable**. The Insert Notification Variable to the Field dialog box appears.

- c From the **Variable Name** field select a variable. The content of the variable is displayed in the Preview field. The **Description** field briefly describes the variable.
- d Click OK.



TIP After you have inserted variables you can add freeform text anywhere in the field. You can also cut and paste variables if you want to rearrange their order.

- 8 For **Message** type the message for this notification event. If you do not specify a message the template message is used. Add pre-defined variables and HTML formatting tags if required.
- 9 Click OK. The notification event is added to the build configuration and displayed in the content area.

Editing Notification Events

Purpose Follow this procedure to edit an existing notification event.



NOTE If you do not specify a subject, message, or notification template the notification event is ignored.

To edit a notification event:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Subscriptions. In the navigation or content areas click the Dimensions project containing the notification event that you want to edit.
- 2 Select the check box next to the notification event and click Edit.



The Edit Notification Event dialog box appears.

- 3 From the **Project** list change the Dimensions project for the notification event.
- 4 From the **Build Configuration** list change the build configuration for the notification event.
- 5 From the **Targets** list change the target to which the notification event will apply. To trigger a notification for all targets select *All Targets*.
- 6 From the **Event** list change the type of event that triggers the notification event. Some events also require that you select an additional option. For example, if you select the *Build is finished* event type you must also select an option from the Build Result list.
- 7 From the **Template** list optionally select a notification template that will be used to populate the subject and message fields in the e-mail message. If you do not want to use a template select *<none>*.

- 8 For **Subject** modify the subject of this notification event. If you do not specify a subject the template subject is used. To add a pre-defined variable do the following:
 - a Place the cursor where you want to add the variable.
 - b Click **Insert Variable**. The Insert Notification Variable to the Field dialog box appears.
 - c From the **Variable Name** field select a variable. The content of the variable is displayed in the Preview field. The **Description** field briefly describes the variable.
 - d Click OK. The variable is added to the end of the field.



TIP After you have inserted variables you can add freeform text anywhere in the field. You can also cut and paste variables if you want to rearrange their order.

- 9 For **Message** modify the message for this notification event. If you do not specify a message the template message is used. Add pre-defined variables and HTML formatting tags if required.
- 10 Click OK. The notification event details are updated in the Dimensions project content area.

Deleting Notification Events

Purpose Follow this procedure to delete notification events.

To delete notification events:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications and click Subscriptions.
- 2 In the navigation or content areas click the Dimensions project containing the notification events that you want to delete.
- 3 In the content areas select the check box next to each notification event that you want to delete.
- 4 Click Delete. To confirm that you want to delete the notification events click Yes.

Adding Notification Subscriptions

Purpose Follow this procedure to add a new notification subscriber to an existing notification event. You can add Dimensions users or groups of users that have been defined in your process model. For information about adding users and groups see the *Process Modeling Guide* or the administration console help.

To add a notification subscription:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications, click Subscriptions, and click the Dimensions project containing the build configuration where you want to add a notification subscription.
- 2 In the navigation area click the build configuration, click Events, and select the notification event where you want to add a notification subscription.

- 3 In the Notification Subscribers section of the content area click New Object.



The Add New Notification Subscriber dialog box appears.

- 4 For **Subscriber Type** select User or Group.
- 5 From the **Subscriber Name** list select a user or group.
- 6 For **Subject** type the subject of this notification subscription. If you do not specify a subject the notification event subject is used. To add a pre-defined variable do the following:
 - a Place the cursor where you want to add the variable.
 - b Click **Insert Variable**. The Insert Notification Variable to the Field dialog box appears.
 - c From the **Variable Name** field select a variable. The content of the variable is displayed in the Preview field. The **Description** field briefly describes the variable.
 - d Click OK. The variable is added to the end of the field.



TIP After you have inserted variables you can add freeform text anywhere in the field. You can also cut and paste variables if you want to rearrange their order.

- 7 For **Message** type the message for this notification subscription. If you do not specify a message the notification event message is used. Add pre-defined variables and HTML formatting tags if required.
- 8 Click OK. The new notification subscription is added to the content area.

Editing Notification Subscriptions

Purpose Follow this procedure to edit an existing notification subscriber for a notification event. You can add Dimensions users or groups that have been defined in your process model. For information about adding users and groups see the *Process Modeling Guide* or the administration console help.

To edit a notification subscription:

- 1 In Dimensions Build click the Notifications tab. In the navigation area click Notifications, click Subscriptions, and click the Dimensions project containing the build configuration where you want to modify an existing notification subscription.
- 2 In the navigation area click the build configuration, click Events, and select the notification event containing the notification subscription that you want to edit. In the

In the Notification Subscribers section of the content area select the check box next to the notification subscription and click Edit.



The Edit Notification Subscriber dialog box appears.

- 3** For **Subscriber Type** select User or Group.
- 4** From the **Subscriber Name** list select a user or group.
- 5** For **Subject** modify the subject of this notification subscription. If you do not specify a subject the notification event subject is used. To add a pre-defined variable do the following:
 - a** Place the cursor where you want to add the variable.
 - b** Click **Insert Variable**. The Insert Notification Variable to the Field dialog box appears.
 - c** From the **Variable Name** field select a variable. The content of the variable is displayed in the Preview field. The **Description** field briefly describes the variable.
 - d** Click OK. The variable is added to the end of the field.



TIP After you have inserted variables you can add freeform text anywhere in the field. You can also cut and paste variables if you want to rearrange their order.

- 6** For **Message** modify the message for this notification subscription. If you do not specify a message the notification event message is used. Add pre-defined variables and HTML formatting tags if required.
- 7** Click OK. The details of the notification subscription are updated in the content area.

Deleting Notification Subscriptions

Purpose Follow this procedure to delete notification subscriptions.

To delete notification subscriptions:

- 1** In Dimensions Build select the Notifications tab.
- 2** In the navigation area select Notifications, select Subscriptions, and select the Dimensions project containing the build configuration where you want to delete notification subscriptions.
- 3** In the navigation area select the build configuration and select Events.
- 4** In the content areas select the check box next to each notification event that you want to delete.
- 5** Click Delete. To confirm that you want to delete the notification events click Yes.

Part 4

Integrating Dimensions Build with Third Party Build Engines

Part 4: Integrating Dimensions Build with Third Party Build Engines contains the following chapters

Using Dimensions with Ant	293
Using Dimensions Build with Openmake	301

Chapter 15

Using Dimensions with Ant

About Using Dimensions Build with Ant	294
Overview	294
Deciding Where to Store the Ant Buildfile	294
Selecting or Creating a Dimensions Project	295
Creating or Selecting Build Areas	295
Importing Ant Buildfiles into a Build Configuration	295
Editing Details of the Imported Targets	297
Editing the Transition Script of an Imported Ant Target	298
Running the Imported Build Job	299

About Using Dimensions Build with Ant

It is possible to integrate your Ant build jobs with Dimensions.

The integration between Dimensions and Ant enables you to:

- Import Ant buildfiles (`build.xml`) files into Dimensions
- Manage your build configurations in Dimensions
- Execute Ant tasks from Dimensions

Overview

Here is an overview of the steps to import your Ant buildfiles to Dimensions.

- 1** Decide whether or not you want to store your Ant buildfile in Dimensions as a controlled file.
- 2** Decide which Dimensions project should contain your Ant build configuration, or create a new project.
- 3** In Dimensions, create any work areas or deployment areas needed to duplicate the directory structure used in your Ant build configuration.
- 4** In Dimensions, create a new build configuration to receive the imported Ant build targets.
- 5** Use the Import Wizard to import the Ant buildfile.
- 6** Edit the imported targets as necessary.
- 7** Check in the build configuration.
- 8** Check in the Ant buildfile.

Deciding Where to Store the Ant Buildfile

The Ant buildfile is required to run the Ant build jobs, so you should decide whether or not to store your Ant buildfile in Dimensions before beginning the import process.

For an informal set of build tasks, you have the option of keeping the Ant buildfile in the build area. For any work that requires version and access control, you should use Dimensions to make the buildfile a controlled file.

Selecting or Creating a Dimensions Project

It is necessary to have a Dimensions project as Dimensions will require this for the creation of a build configuration. You should select or create a Dimensions project to receive the targets defined in the Ant buildfile.

If you do not already have source files archived in Dimensions, use the structure defined in the Ant buildfile to set up your source directories under the project.

See the *Serena Dimensions CM User's Guide* for instructions on creating a Dimensions project.

Creating or Selecting Build Areas

You should have a build area ready before importing the Ant buildfile. The base directory of the Ant buildfile should correspond to a Dimensions work area.

Also consider using Dimensions deployment areas for some or all of the directories. For example, if the buildfile refers to a \dist directory, that would be a good candidate for a deployment area.

If it is necessary to create a new build area, remember to attach the work area. The procedure for doing this is described in [Build Areas](#) on page 191.

Importing Ant Buildfiles into a Build Configuration

Purpose Follow this procedure to import an Ant buildfile into Dimensions. After the buildfile is successfully imported, the Ant targets are added to the build configuration.

To import an Ant buildfile into a build configuration:

- 1 In Dimensions Build, select Build Management.
- 2 Create a new build configuration to receive the imported Ant buildfile.
- 3 In the content area, in the Build Targets section, click Launch Import Wizard.



The Import Wizard appears. On the left side of the wizard there are links to four pages:

- Select Import Type
- Choose File

- Parameters
- Confirm

By default, the Select Import Type page is displayed.

4 On the Select Import Type page, select **Ant build configuration (build.xml) file**.



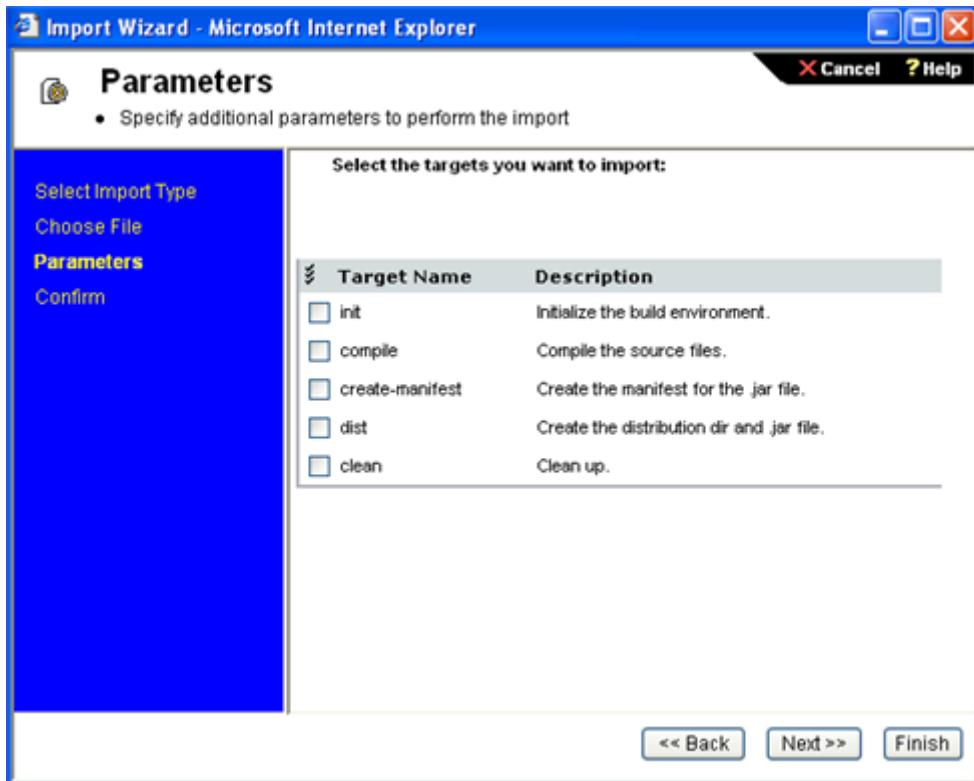
NOTE An Ant build configuration file need not be named build.xml, of course, but for the sake of convenience, the prompt uses the name build.xml.

Click Next. The Choose File wizard page appears.

5 On the Choose File page, in the **Select Ant build configuration file** field, do one of the following:

- Type the full path to the Ant buildfile that you want to import.
- Click Browse to navigate to the Ant buildfile that you want to import.

Click Next. The Parameters page appears.

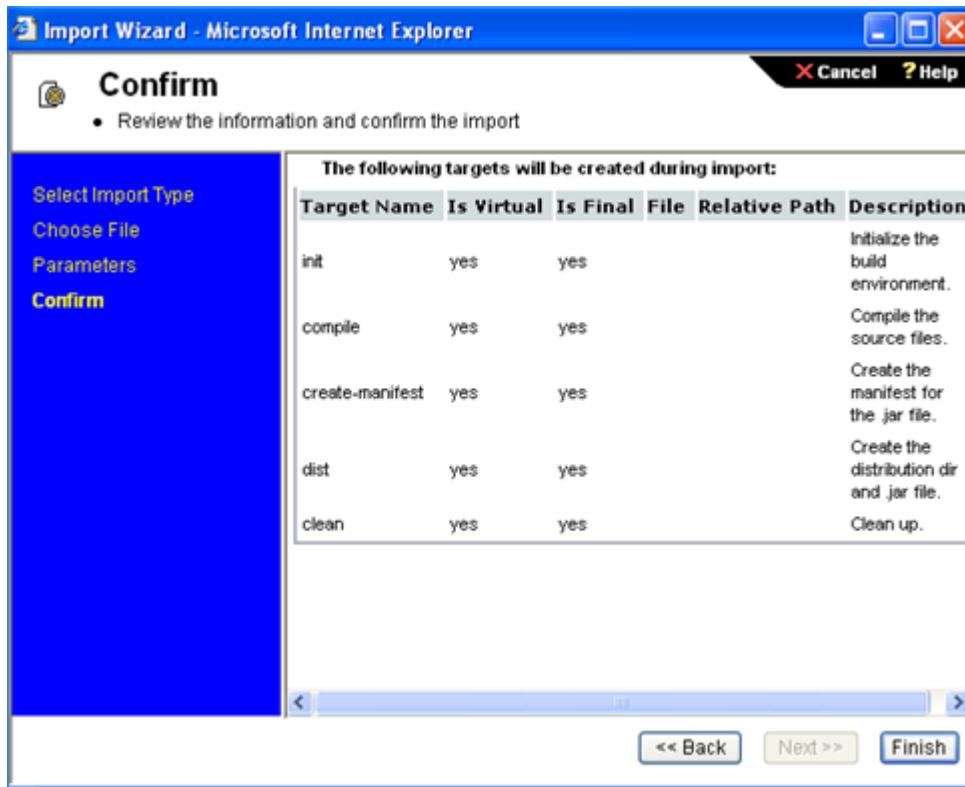


6 The **Parameters** page displays the following information:

- The name of each target found in the Ant buildfile.
- The description given in the Ant buildfile. If no descriptions are entered in the buildfile, none will appear in the wizard page.

7 Select the targets that you want to import. Use the triple check mark to select all targets.

- 8** Click Next. The Confirm page appears.



- 9** On the Confirm page review the details of the targets that you are going to create.



NOTE If there are details you wish to edit, you will be able to do that from within Dimensions.

- 10** Click Finish. The dialog box displays a brief confirmation, and then the imported targets appear in the content area.
- 11** Check in the build configuration to preserve the imported targets.

Editing Details of the Imported Targets

Purpose Follow this procedure to edit details of targets imported from an Ant buildfile. For example, you may wish to edit small details such as the description. You may also want to inspect the transition script, as each target will still depend on the Ant buildfile.

To edit details of targets imported from an Ant buildfile:

- 1 Select the Build Management tab if it is not selected already.
- 2 Expand the Dimensions Projects tree until the tree area shows the build configuration to which the Ant targets were imported.
- 3 Select the build configuration. You should be able to see the imported targets. Verify that the new targets were added to your list of build targets.

- 4 Check out the build configuration if it is not checked out already.
- 5 Click the name of the target you wish to edit. The Edit Build Target dialog box appears.
- 6 Edit the target details as desired. Click OK to accept the dialog box.
- 7 Check in the modified build configuration.



NOTE Any changes to the name or description of the target will display only in Dimensions and will not affect the Ant buildfile itself.

Editing the Transition Script of an Imported Ant Target

Purpose Follow this procedure to edit the transition script for a target imported from an Ant buildfile. Note that this is not the same as editing the Ant buildfile itself, and that the buildfile is still needed to build the targets that were imported from it.

To edit the transition script of a target imported from an Ant buildfile:

- 1 Select the Build Management tab if it is not selected already.
- 2 Expand the Dimensions Projects tree until the navigation area shows the build configuration to which the Ant targets were imported.
- 3 Select the desired build configuration.
- 4 Check out the build configuration if it is not checked out already.
- 5 Expand the build configuration until the individual targets are visible in the navigation area. You cannot edit the transition script unless you can see the individual targets in the Dimensions Projects tree.
- 6 Select the target whose transition script you wish to edit. The Transition Details section becomes visible in the content area.



- 7 In the content areas, in the Transition Details section, click Edit Build Script.



The Edit Transition Script dialog box appears. The Script Content field displays the transition script. For example:

```
ant -buildfile "build.xml" "create-manifest"
```

- 8 Edit the transition script as desired.



NOTE Be careful editing the transition script. If you break the connection between the target and the Ant buildfile, Dimensions will lose the ability to build the target.

- 9 Check in the build configuration to save the edited target.

Running the Imported Build Job

To run the build configuration, you should follow the procedures described in [Executing Builds in the Build Administrator](#) on page 227. You should also pay attention to the following:

- Make sure the Ant buildfile is in the build directory or is in a location that both Ant and Dimensions can find.

If you have targets that need to be preserved, create the Dimensions archives for those targets and then be sure that the **Do not preserve targets** checkbox is not selected in the Run Build wizard.

Chapter 16

Using Dimensions Build with Openmake

About the Integration with Openmake	302
Templates	302
Integration Steps	303
Openmake Build Configuration Type	304
Search Paths	304
Importing TGT Files into a Build Configuration	307
Producing Impact Analysis Data	309
Return Codes	310

About the Integration with Openmake



IMPORTANT!

- Serena ChangeMan Builder is based on Openmake from Catalyst Systems; the information below refers to both Serena ChangeMan Builder and Openmake.
- Only Openmake version 6.4.1 has been tested with Dimensions Build.
- If you have previously configured build jobs in either Serena ChangeMan Builder or Openmake, you can import and compile those build jobs in Dimensions Build. For details see "["Importing TGT Files into a Build Configuration" on page 307](#)".

The integration between Serena ChangeMan Builder and Dimensions Build enables you to:

- Import Openmake targets into Dimensions Build.
- Manage your build configurations in Dimensions Build.
- Compile your targets from Dimensions Build using the Openmake engine.

For details about using and configuring Serena ChangeMan Builder see the accompanying documentation from Catalyst Systems.

Templates

A template is a customizable text file containing variables and control words. In Dimensions, templates are used to execute builds and execute general commands via remote job execution. For details about the templating language, and the Openmake build templates and variables, see the *Developer's Reference*.

Integration Steps

To integrate Openmake with Dimensions Build follow these steps:

Step	Description	Additional Information
Create a build configuration	In Dimensions Build create a build configuration, specify the Openmake build configuration type, and verify that the project name that you specify matches the project name defined on the Openmake KB server.	Chapter 8, "Managing Build Configurations" on page 183 and "Build Configuration Types" on page 152
Edit the Openmake build options	In Dimensions Build edit the Openmake options in the build configuration as required. The option DMSEARCHPATH should contain the name of the Openmake search path.	Chapter 8, "Managing Build Configurations" on page 183
Set up search paths in Openmake	In the Openmake web client set up search paths as normal.	ChangeMan Builder documentation
Set up TGT files in Openmake	In the Openmake web or ISPF clients set up TGT files as normal.	ChangeMan Builder documentation
Verify the search paths	<p>In Dimensions Build verify that the search path exactly matches the search path defined in the Openmake KB server for the current project and phase (DEV, UT, ST, etc.). The search path is derived from the relationship of the deployment areas to the stages in the Global Stage Lifecycle. To setup search paths do the following:</p> <ul style="list-style-type: none"> a Set up a Dimensions lifecycle or use an existing one. b Add stages to the Global Stage Lifecycle or use the default stages. c Define deployment areas for each stage in the Global Stage Lifecycle. 	Lifecycle Management chapter and File Area Definitions chapter in the Process Modeling User's Guide
Import TGT files	In Dimensions Build use the import target wizard to import your TGT files into your build configurations.	"Importing TGT Files into a Build Configuration" on page 307
Produce impact analysis data	In Dimensions Build optionally specify a build template variable that triggers the production of impact analysis data during a build.	"Producing Impact Analysis Data" on page 309
Run the build	Run your build from Dimensions Build or from a Dimensions client.	Chapter 9, "Executing Builds in the Build Administrator" on page 227
Monitor the build	<p>In Dimensions Build monitor the initial progress of your build. You can also use Dimensions Build to view a Bill of Materials.</p> <p>If you are running a build on a Windows or UNIX machine a web browser opens automatically and displays information from the Openmake knowledge base server about the progress of the build.</p>	Chapter 13, "Monitoring Builds in the Build Administrator" on page 261

Openmake Build Configuration Type

In Dimensions Build a build configuration type specifies the options that are unique to a particular build configuration and are required to run a build. Dimensions Build includes a build configuration type called *Openmake* that has a number of pre-defined variables. You can configure these options to suit your build environment. The variables are inputs to Openmake build templates. For details about the templating language and the Openmake build templates, see the *Developer's Reference*.

If you select the *Openmake* type when you create a new build configuration, Dimensions Build automatically populates the build configuration and the build areas with the build options required to perform an Openmake build. After you have created the build configuration you can review and edit the build options; for example, you can change the location of an Openmake installation on a build node.

- For details about managing the default build configuration type options see "[Build Configuration Types](#)" on page 152.
- For details about creating build configurations see [Chapter 8, "Managing Build Configurations"](#) on page 183.
- For details about managing build options defined for build configurations see "[About Build Options](#)" on page 219.

Search Paths

You normally build projects at different release levels in the development lifecycle, such as DEVELOPMENT, UNIT TEST, SYSTEM TEST, and RELEASE. In Openmake the level of code is managed by source code search paths.

A search path is a collection of directories (on z/OS, library data set name prefixes), listed in the order required to create a specific build level. A build looks for source code in the directories listed in the search path, in the order in which the directories are listed. When a piece of source code is found, the search for that code stops.

The relationship between stages in the Dimensions global stage lifecycle and Openmake search paths is critical. The names of the stages and the search paths *must* be the same, and the search paths must match the stage sequence defined in the Dimensions global stage lifecycle. For example, if your global stage lifecycle is as follows:

DEVELOPMENT
RELEASE
SYSTEM TEST
UNIT TEST

you need to set up search paths in Openmake such as the following:

C:\MyDevSearchPath\DEVELOPMENT
C:\MyDevSearchPath\RELEASE
C:\MyDevSearchPath\SYSTEM TEST
C:\MyDevSearchPath\UNIT TEST

Additionally, you must verify that the deployment areas associated with the stages in the global stage lifecycle contain the same directory names that you specified in the Openmake search path.

Search Path Concatenation

When you define the search path directories in a specific order, the paths are concatenated so that items that are not in the lower level build areas are obtained from the higher level build areas. For example, a build at the SYSTEM TEST stage looks for source items from the SYSTEM TEST and higher level RELEASE build areas, but any changes in source items in the lower level DEVELOPMENT and UNIT TEST build areas are ignored. A build always compiles the active item revisions for the stage where the build is executed.

For example, assume that you have the following build stages and items:

Build stages	Items and revision numbers		
UNIT TEST	foo.c-3		
SYSTEM TEST	foo.c-2	boo.c-2	
RELEASE	foo.c-1	boo.c-1	goo.c-1

If you build at the RELEASE stage, foo.c-1, boo.c-1 and goo.c-1 are compiled:

Build stages	Items and revision numbers		
UNIT TEST	foo.c-3		
SYSTEM TEST	foo.c-2	boo.c-2	
RELEASE	foo.c-1	boo.c-1	goo.c-1

If you build at the SYSTEM TEST stage, foo.c-2, boo.c-2 and goo.c-1 are compiled:

Build stages	Items and revision numbers		
UNIT TEST	foo.c-3		
SYSTEM TEST	foo.c-2	boo.c-2	
RELEASE	foo.c-1	boo.c-1	goo.c-1

If you build at the UNIT TEST stage, foo.c-3, boo.c-2 and goo.c-1 are compiled:

Build stages	Items and revision numbers		
UNIT TEST	foo.c-3		
SYSTEM TEST	foo.c-2	boo.c-2	
RELEASE	foo.c-1	boo.c-1	goo.c-1

Active Item Revisions

Active item revisions prevent situations where the latest revision at a build stage is *not* the latest revision further along a search path. For example, assume that you have two revisions of the same item, foo.c-2 and foo.c-3. You action both item revisions to the UNIT TEST build stage, and then action foo.c-3 only to the SYSTEM TEST build stage. foo.c-3 is the latest revision of the item and is the active item revision at both the UNIT TEST and SYSTEM TEST build stages. Because foo.c-2 is an earlier revision of the item it is inactive at the UNIT TEST build stage and is not exported to UNIT TEST build areas.

Both item revisions at UNIT TEST (active item shown in bold):

Build stages	Items and revision numbers	
UNIT TEST	foo.c-2	foo.c-3

foo.c-3 actioned to SYSTEM TEST (active item shown in bold):

Build stages	Items and revision numbers	
UNIT TEST	foo.c-2	
SYSTEM TEST		foo.c-3

Additional Search Locations

The search path may also include additional locations not defined as Dimensions build areas, for example:

- Directories or libraries (PDS/E or PDS) containing third party source code.
- Object modules that are not kept under Dimensions control.
- Directories containing standard *include* and library files.

The additional locations are defined in the Dimensions Build project.

The compiler and linker build tools look for the sources in the search path environment.

Importing TGT Files into a Build Configuration

Purpose Follow this procedure to import an Openmake TGT file into a build configuration in Dimensions Build. After the TGT file is successfully imported, the target and its dependencies are added to the build configuration.

To import an Openmake TGT file into a build configuration:

- 1 In Dimensions Build select the Build Management tab.
- 2 In the navigation area select the build configuration where you want to import the TGT file, and click Build Targets.



NOTE The build configuration must be checked out.

- 3 In the content area, on the Build Targets tab, click Launch Import Wizard.

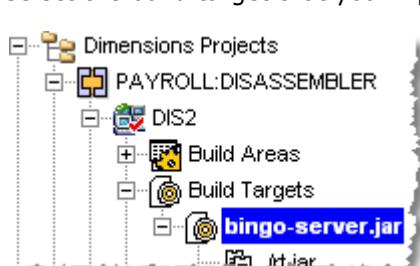


The Import Wizard appears. On the left side of the wizard there are links to four pages: Select Import Type, Choose File, Parameters, and Confirm. By default, the Select Import Type page is displayed.

- 4 On the Select Import Type page, select Openmake TGT file and click Next.
- 5 On the Choose File page, in the **Select OpenMake TGT file** field, do one of the following:
 - Type the full path to the TGT file that you want to import.
 - Click Browse, navigate to the TGT file that you want to import, select it, and click Open.
- Click Next.
- 6 The **Parameters** page displays the following information:
 - The name of the target file.
 - The relative path to the directory where the target is to be built.
 - The sources required to build the target.
 - The relative paths of the sources in relationship to the target.

(Optional) Modify the target name and its relative path.
- 7 Select the sources that you want to import from the TGT file.

- 8** Click Next.
- 9** On the Confirm page review the details of the target that you are going to create and click Finish.
- 10** In the content area of your build configuration click the Build Targets tab. Verify that the new target was added to your list of build targets.
- 11** Click the Sources tab. Verify that the new sources were added to your list of build sources.
- 12** To review the transition script that will be used to build the target do the following:
 - a** In the navigation area expand your build configuration, click Build Targets, and select the build target that you imported.



- b** In the content areas, in the Transition Details section, click Edit Build Script.



The Edit Transition Script dialog box appears. The Script Content field displays the following default Openmake template:

om_build.template



IMPORTANT! The default Openmake template is located in the directory <install root>/templates. If you customize the template, Serena recommends that you place it under source control and in the build directory. The Openmake build templates that are supplied with Dimensions CM are described in the *Developer's Reference*.

- c** Click OK.
- 13** Optionally add build options and option groups to the build target.

Producing Impact Analysis Data

Purpose Serena ChangeMan Builder enables you to specify a build template variable that triggers the production of impact analysis data during a build. You can add the variable to the build dialog boxes in Dimensions clients or to the Run Build wizard in Dimensions Build. The output is placed in the following file:

MVS: <build area root>.XML(IMPACT)

Windows: <build area root>\impact.xml

UNIX: <build area root>/impact.xml

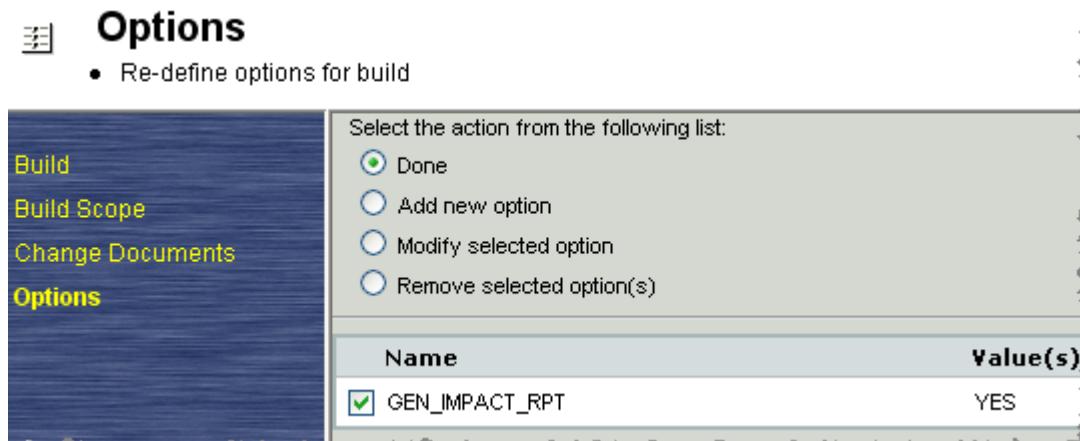
To produce impact analysis data (desktop, web, and ISPF clients):

In a build dialog box, in the **Build Options** field, type GEN_IMPACT_RPT=YES.

For information about running builds from these clients see the *User's Guide* and the *Dimensions for z/OS User's and Administrator's Guide*.

To produce impact analysis data (Dimensions Build):

- 1 Open the Run Build wizard and specify parameters as described in "Executing Builds in the Build Administrator" on page 227.
- 2 In the Options page select **Add new option** and click Next.
- 3 For **Name** type GEN_IMPACT_RPT.
- 4 For **Value** type YES.
- 5 Click Next.
- 6 In the Options page select the build template variable that you have just defined.



- 7 Click Run.

Return Codes

Dimensions Build returns the following Openmake-related return codes:

- Distributed builds: returns RC=0 if all build templates completed successfully. The Openmake build template will return a non-zero value if any step in the Openmake processing returned a non-zero value. Openmake returns a non-zero value if any language processor failed.
- MVS builds: the return code from an Openmake step is zero if Openmake system level processing is successful. If a language processor or other sub-step fails due to a compile or link error, this failure is not reported via this return code but is reported in the response file associated with the completion of the build. The build server must replace the job's return status with the parsed step return status.

Part 5

Appendices

Part 5: Appendices contains the following chapters

Troubleshooting	313
The Dimensions Build Architecture	319
Dimensions Build Utility Programs	323
Dimensions Build Security	339

Appendix A

Troubleshooting

About Troubleshooting	314
Viewing Errors Listed in Build Monitor Events	314
Specific Errors	315
Using Temporary Files to Debug Your Builds	317

About Troubleshooting

This section describes various conditions and error messages you may see in the course of using Dimensions. Later, the section also covers debugging information you can find in temporary files.

Error messages may be written to any of several areas:

- in Build Execution Statuses
- in Build Monitor Events
- in the Tomcat window
- in wizard pages
- On MVS, in SDSF/JES spool
- On MVS, in the MVS System log

If you are having a problem, you should look for error messages, and then search within this chapter for any information relevant to the messages you see.

If you are unable to find helpful information in this chapter, you can contact Serena customer support at support.serena.com.

Viewing Errors Listed in Build Monitor Events

If your build job encounters an error during execution, Dimensions displays an error listing in Build Monitor Events:

Build Monitor Events						
Order	Step	Type	Time	Target	Task	Message
1		Build started	Fri Aug-04 3:58PM			
2		Target Build started	Fri Aug-04 3:58PM	hello.exe		
3		Target Build finished	Fri Aug-04 3:58PM	hello.exe		rc=1
4		Build finished	Fri Aug-04 3:58PM			

Recall that you view Build Monitor Events in one of two ways:

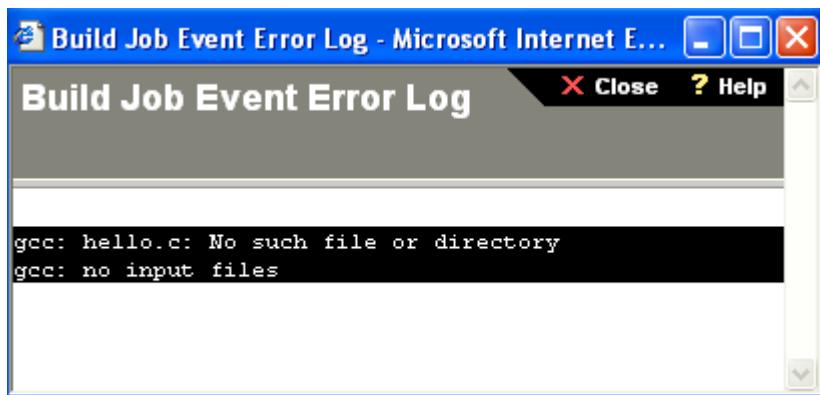
- After a build executes, the Build Execution Statuses dialog box appears; click the See Build Events icon on that dialog box to view Build Monitor Events.
- From the Build Job Monitoring tab, locate your build job under the History folder, then click the appropriate link under Execution History to view Build Monitor Events.

To view the error log:

- 1 Click the **View error log** icon:



The Build Job Event Error Log appears:



- 2 Review the information in the error log to determine what went wrong.

Specific Errors

This section presents a list of specific errors you may see, with possible causes.

Area <area name> is in use

The full error message is:

Area <area_name> is in use and you cannot update <property> at this time

If you see this error message, it means that you are trying to edit a build area that has already been associated with a checked-in build configuration. Once a build configuration with an attached build area has been checked in, you can no longer modify the following build area properties:

- Area ID
- Stage ID
- Network node
- Directory

Attempt to close invalid connection

If your build does not execute, and if you see this message in the Tomcat window, it may mean that Tomcat has timed out your connection. Restart Tomcat using Start > Programs > Serena > Common Tools > Start Common Tomcat.

Failed to authenticate to the build agent

If your build displays this error message, it may mean that the you have entered an incorrect password for the user ID under which Dimensions logs in.

It could also mean that the build agent is not installed or not running on the build machine.

Failed to find product-specific upload rules

If your build executes but displays this error message, it may mean that your Dimensions administrator has not yet defined a default item library. Dimensions will not be able to upload the build target to Dimensions as a new item unless it can locate a default item library.

No build areas found for the selected configuration

If you see this error in the Run Build wizard, it could mean that you are trying to use the Build Scheduling tab to execute a never-before-executed version of a build configuration.

If a version of a build configuration has been checked in, but has never been run, it will not contain a build area association. A version of a build must be run at least once before you can execute it from the Build Scheduling tab.

If you see this error when trying to execute the build from the Build Management tab, it means that the build configuration has no build areas attached.

Template pbem_stop.cmd open failed

If your build job displays this error message, it may mean that the temp directory defined in the Dimensions configuration file contains a space.

Edit `<DM_ROOT>/dm.cfg` and change the value of DM_TMP to a path that does not contain spaces.

You do not have a role to extract item

If your build job displays this error message, it may mean that the Dimensions user you have logged in as does not have a role that allows updating the build target. More specifically, the Dimensions user may lack a role on the first transition of the item. For example, an .EXE file using the default LC_EXECUTABLE lifecycle will transition from BUILT to TESTED. If the Dimensions user does not have a role on that transition, the check-in of the build target may fail.

Use the Dimensions Administration Console to check the appropriate roles and transitions, or contact your Dimensions administrator.

Using Temporary Files to Debug Your Builds

During the course of a build, Dimensions produces temporary files. These temporary files contain information that can help you debug your build problems.

Normally these temporary files are deleted after each build, but you can cause them to be preserved.

Preserving Temporary Files

This procedure will enable you to preserve the temporary files that Dimensions produces.



NOTE You must have access to the Tomcat server used by Dimensions. After this procedure, you will need to restart Tomcat.

To preserve temporary files:

- 1 On the Dimensions server, navigate to the Dimensions install directory (DM_ROOT).
- 2 Navigate further to DM_ROOT\Common Tools\tomcat\<version #>\webapps\bws.
- 3 Open the file bws\WEB-INF\web.xml.
- 4 Search for the string "clean". You should see the following section of the file:

```
<init-param>
    <param-name>clean.temp.files</param-name>
    <param-value>true</param-value>
    <description>
        Whether to clean temporary files on remote hosts. Default -
        true.
    </description>
</init-param>
```

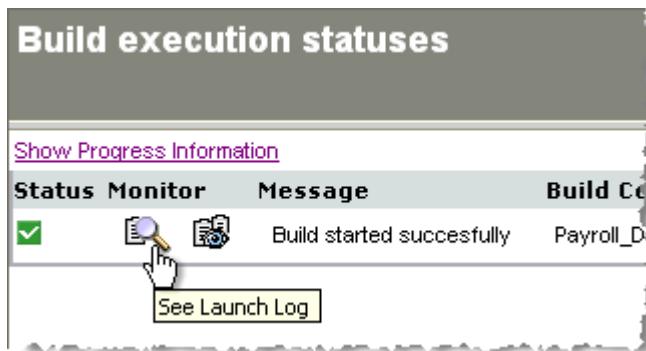
- 5 Change the param-value to FALSE.
- 6 Save the file, and restart Tomcat. The temporary files will now be preserved.

Locating the Temporary Files

After you have modified the web.xml file to preserve temporary files, and restarted Tomcat, the next build you run will preserve its temporary files. To locate the temporary files, use this procedure.

To find the temporary files:

- After running a build, the Build Execution Statuses dialog box should be visible. Click the See Launch Log icon:



The View Launch Attempt Log dialog box appears.

- Scroll to the bottom of the Launch Attempt Log. You will see something like this:

```
Passing the template for execution
cmd = cmd /c C:\DOCUME~1\dmsys\LOCALS~1\Temp\TPL1B6~3.BAT
      >"C:\DOCUME~1\dmsys\LOCALS~1\Temp\tpl2b64-2.tmp" 2>&1
Template processing and submission complete: no errors
```

- Note the location of the .BAT and .TMP files. This location is controlled by the DM_TMP variable. Navigate to this directory.



NOTE On Windows, the directory Documents and Settings\<username>\Local Settings is a hidden directory. You will have to show hidden files and folders to see it.

- Locate the files with file extensions of .BAT or .TMP. (The exact names of the files are different each time.) These are the files with useful information. You may need to compare the files produced with similar files from an example build that works.

The BRD File

Another file you will see listed in the Launch Attempt Log is the BRD file. For example:

```
BRD file name = C:\DOCUME~1\dmsys\LOCALS~1\Temp\ptb641.tmp
```

This file contains information about what the values of build options are, about the steps in the build, and the order in which the steps occurred.

In particular, users creating Openmake-type builds can read the values of the Openmake build variables from this file.

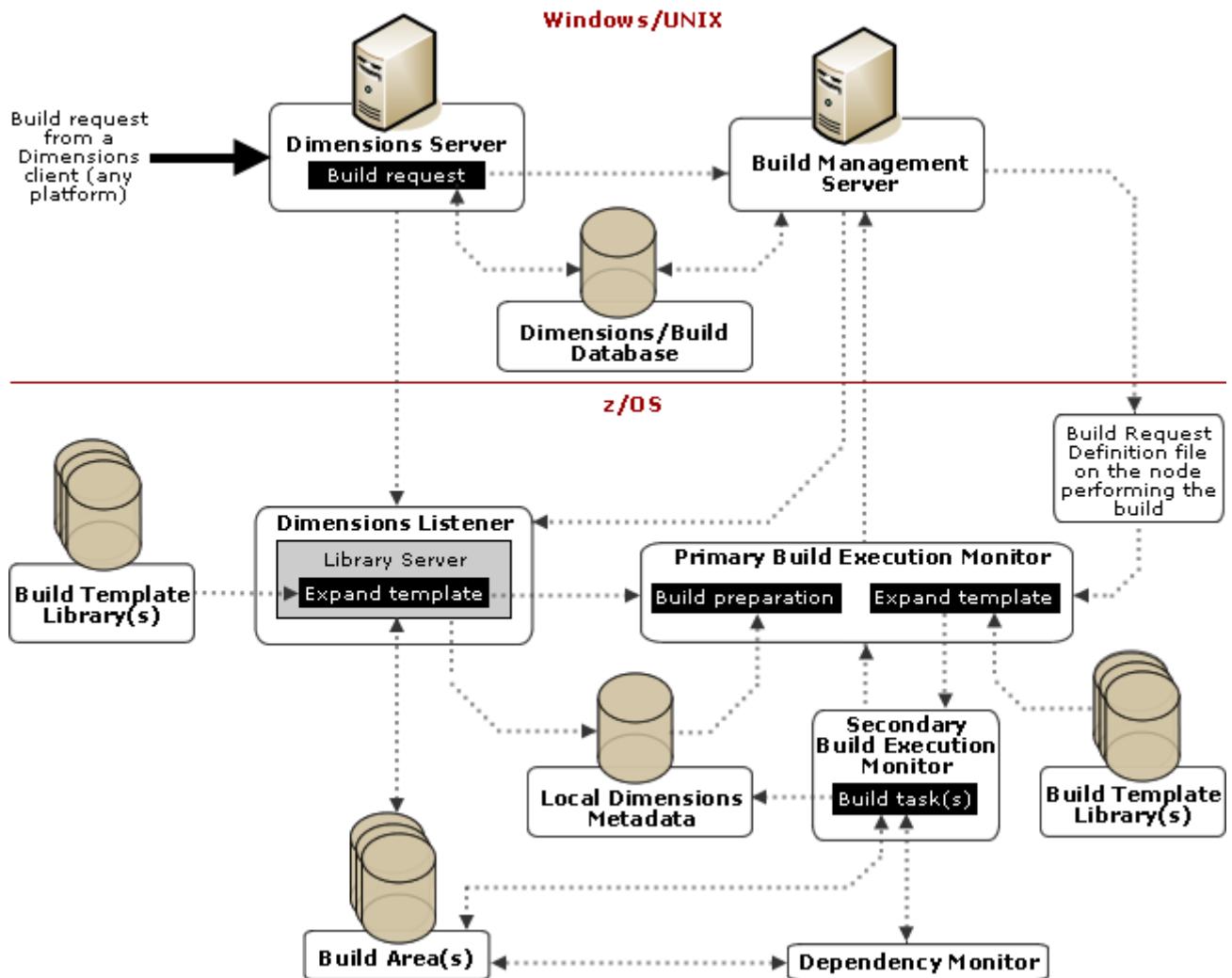
Appendix B

The Dimensions Build Architecture

This appendix describes the architecture of Dimensions Build on MVS (mainframe) and distributed platforms.

MVS Platforms

The diagram below illustrates the architecture of Dimensions Build on MVS platforms:



Pre-requisites: A properly configured build configuration.

The build process begins when a build request is initiated from a Dimensions client. The Dimensions server sends the build request and any necessary information to the Build Management Server. The Build Management Server then does the following:

- Gathers all the information about the build being requested (targets, script files, build options, etc.) and places it in a Build Request Definition (BRD) file on the node performing the build.
- Contacts the Dimensions Listener/Agent running on the build node and passes it the build request for processing.

To handle the build request the Dimensions Listener/Agent starts the Primary Build Execution Monitor (PBEM). The PBEM is started via a Dimensions template therefore its start up parameters can be modified as necessary. The main job of the PBEM is to oversee and control the build process from start to finish. Using the BRD file from the Build Management Server, the PBEM invokes sub build processes to perform each step required by the build. For details more about the PBEM see [page 324](#).

For each build step in the BRD the PBEM performs a process called Build Preparation. This process determines whether the given step needs to execute based on the files' local metadata. If the date and time stamps for the sources and targets for a given step match, a build is not required and the build step is skipped.

For each build step that needs to execute, the build template file that is required to build the step is expanded and executed. Most of these templates run in the context of a Secondary Build Execution Monitor (SBEM). The main job of the SBEM is to:

- Oversee and execute a given build step by expanding additional templates (if necessary).
- Respond to any directives that are in the build template.

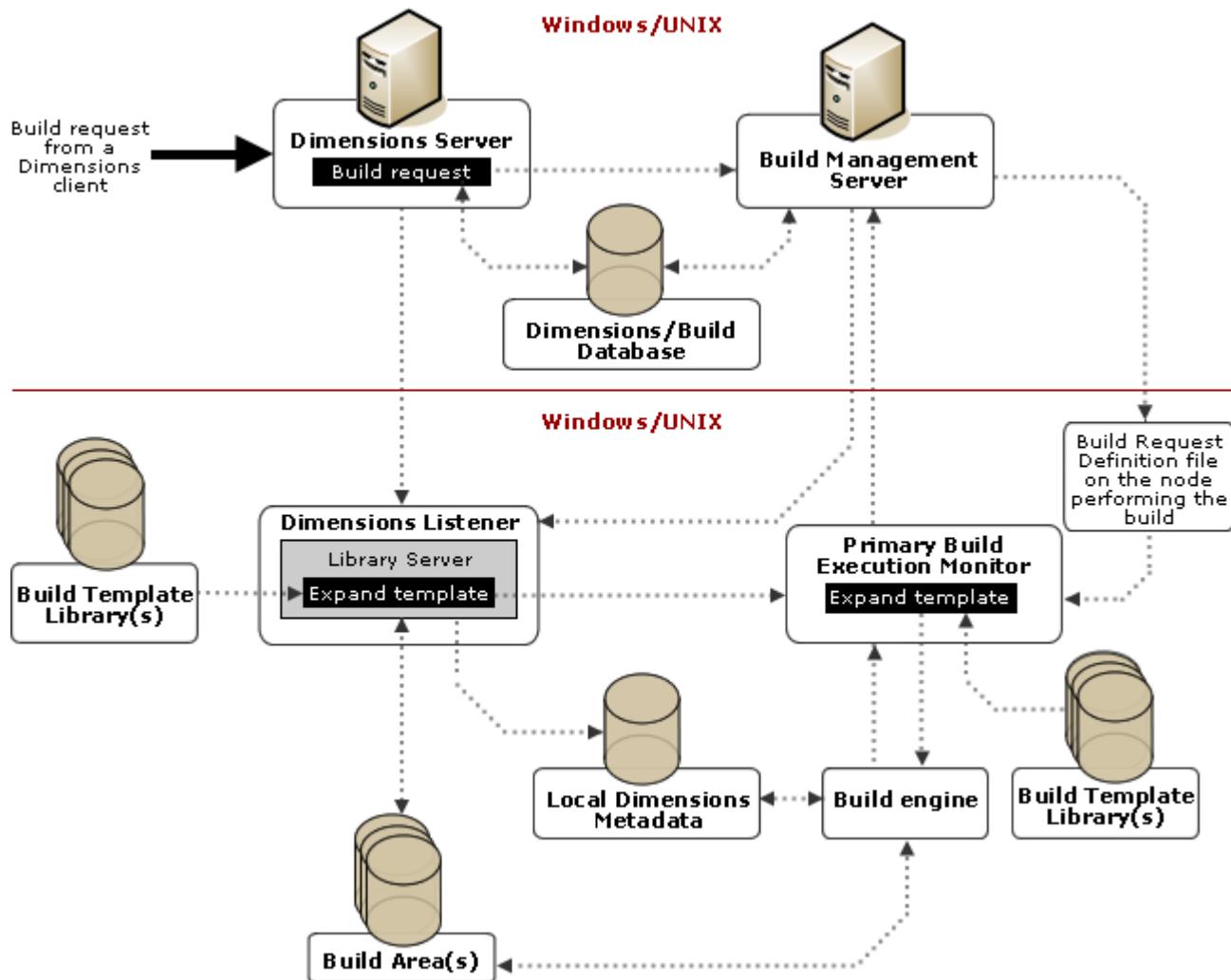
As a build template is prepared for execution, it can initiate the Dependency Monitor to watch specified DD cards for activity and record the results. This information goes in the Bill of Materials (BOM) report to be used later. The build task is then run, outputs are created in the build areas, local metadata is updated, and the final progress and log files are reported back to the PBEM on completion.

As each step completes, the build log and BOM are passed back to the Build Management Server. The server records the progress in the database so that it can be displayed to the user, and parses and stores the BOM information. If output collection is enabled, outputs are collected and new revisions created in Dimensions as necessary.

After the entire build process is completed, the PBEM reports a final status and shuts down. The Build Management Server continues to collect build outputs until the results of all the build steps have been completed.

Distributed Platforms

The diagram below illustrates the architecture of Dimensions Build on distributed platforms:



Pre-requisites: A properly configured build configuration.

The build process begins when a build request is initiated from a Dimensions client. The Dimensions server sends the build request and any necessary information to the Build Management Server. The Build Management Server gathers all the information about the build being requested (targets, script files, build options, etc.) and places it in a Build Request Definition (BRD) file on the node performing the build. The Build Management Server contacts the Dimensions Listener/Agent running on the build node and passes it the build request for processing.

To handle the build request the Dimensions Listener/Agent starts the Primary Build Execution Monitor (PBEM). The PBEM is started via a Dimensions template therefore its start up parameters can be modified as necessary. The main job of the PBEM is to run the build scripts that invoke a third party build tool, including any build options setup in the

configuration. After the third party build tool performs and completes the build, the build scripts invoke utilities that communicate the status of the build back to the PBEM.

After the entire build process is completed, the PBEM reports a final status and shuts down. The Build Management Server continues to collect build outputs until the results of all the build steps have been completed.

Appendix C

Dimensions Build Utility Programs

The Primary Build Execution Monitor	324
The Secondary Build Execution Monitor	327
MDHLLNK0	331
bldcomms	336
Loading Multiple Members	337

The Primary Build Execution Monitor



NOTE For information about the templates and template symbols referred to below, see the *Developer's Reference*.

On mainframe platforms the Primary Batch Execution Monitor (PBEM) is a utility that controls, monitors, and reports results for builds that are initiated by Dimensions Build. On distributed platforms the PBEM run scripts for builds that are initiated by Dimensions Build. When running in a distributed environment the PBEM uses the templater to synchronously start sub tasks.

The PBEM receives input data from Dimensions Build in XML format in a Build Request Definition (BRD) file. The BRD file contains global information and a list of build steps to be executed. The BRD file is arranged as a symbol table, and a collection of sub-symbol tables for each build step. The BRD initiates, and communicates with, sub tasks to execute the build steps. Steps may have dependencies between them; the PBEM is responsible for managing the step dependency analysis and only starts steps when their dependent steps have completed.

The PBEM sends output data to the build server, via a Simple Object Access Protocol (SOAP) utility, when:

- Processing starts.
- A step is cancelled.
- A generalized build step ends.
- A build completes.

The generalized PBEM flow is as follows:

- Reads the BRD file.
- Constructs scripts for executing each individual step.
- Examines targets to decide whether to skip steps based on source dependency analysis.
- Controls the execution of steps and decides whether to execute steps synchronously or asynchronously.
- Coordinates the return of results to the build server.

In normal operations the PBEM is initiated by REXEC processing in the Dimensions listener service, however you can also run it as a stand alone application.

Running the PBEM

To run the PBEM on distributed platforms type pbem at a command prompt. On MVS you can initiate the PBEM in a batch job using the template MDHBPBM0.

PBEM Parameters

Use the following parameters on distributed and mainframe platforms:



IMPORTANT! Apart from the -m and -M parameters, all other parameters are not case-sensitive.

Parameter	Description
-b <brdname>	Specifies the name of the BRD file to be processed.
-d <symbol> <value>	Sets <symbol> to the value that you specify for <value>. Use a pair of double quotation marks ("") to surround values that include spaces. For example: -d DMFOO "last used"
-c <number>	Sets the maximum concurrency allowed for this execution where <number> is a decimal number. 0 or 1 force no concurrency and any template requests for concurrent execution are ignored. If you use a number that exceeds the maximum number of queued connection requests that TCP/IP can accommodate, processing errors will occur.
-q	Causes queues to be dumped at strategic points.
-t <output spec>	Specifies where to deliver the trace output. Use \$ for stdout. On MVS you can use DD:ddname.
-a	Specifies that the default allocator will be used.
-p	Traces parse processing.
-l	Traces director processing.
-v	Prints versions of components.
-M	Traces storage allocations and frees.
-m	Traces message management logic.
-n	Ignores the symbol DMSERVER in a BRD file so that a BRD can be run standalone.
-i	Dumps symbol tables at strategic points.

PBEM Configuration Symbols

The following symbols in the Dimensions configuration file affect the PBEM. The configuration file, dm.cfg, is located in the following directories or data set.

- (Windows): %DM_ROOT%
- (UNIX): \$DM_ROOT
- (MVS): MDH.V1010.MDHPARM(MDHTDCFG)

DMALT SERVER

Enables you to change the address of the build server that is defined in the BRD file, typically the server from where the build is launched, and specify the address of an alternative build server.

Usage scenario: When you launch a build from a virtual machine, and are building on a remote node, the address of the build server in the BRD file on the remote node may be the physical machine hosting the virtual machine, and not the actual virtual machine. You can use DMALT SERVER to change the address to that of the virtual machine.

Must be in the following format:

`http://<IP address of the machine hosting the build server>:<Tomcat port number>/bws/services/monitor`

Example:

`http://<dimensions-dev>:<8080>/bws/services/monitor`

DM_MAX_PBEM_RETRIES

Specifies the maximum number of times the PBEM tries to send a message to the build server. Can be helpful if the build server is heavily loaded.

DM_BUILD_OPTIMIZE

Prevents the build optimizer from performing steps that are not required.

DM_BUILD_OPTIMIZE_METADATA

Ensures that any DEPENDENCY records sent to the PBEM in the BRD file have METADATA records. Enables tests based on an item's file version as well as the timestamps.

DM_BUILD_OPTIMIZE_EARLY_TEST

Performs optimization tests before template expansion, which can improve performance. If you use this symbol in the configuration file it globally enables or disables optimization tests. If you use this symbol in a build configuration it overrides the global setting.

DM_BUILD_OPTIMIZE_DEPENDENCY

Causes the optimizer to request that DEPENDENCY records are added to the BRD, or rebuilds the step. This feature is useful in complex builds that rely on the dependency data to work. However, dependency information is not mandatory so can you turn this symbol off.

Example JCL for Running the PBEM

```
//STEP200 EXEC PGM=MDHLPBEM,
// PARM='POSIX(ON),ENVAR("_CEE_ENVFILE=DD:MDHPDIMV")'
//STEPLIB DD DISP=SHR,DSN=MDH.V1010.MDHLLIB
//          DD DISP=SHR,DSN=MDH.V1010.MDHLLPA
//MDHPTRCE DD SYSOUT=*
//MDHPDIMV DD DISP=SHR,DSN=MDH.DIM671.PARM(MDHTDIMV)
//CEEPRINT DD SYSOUT=*
//MDHPPRNT DD SYSOUT=*
//MDHPBRD DD PATHOPTS=(ORDONLY),
//          PATH='/tmp/my_brd.xml'
//MDHPPARM DD *
-t DD:MDHPTRCE -q -l
-b DD:MDHPBRD
-c 2
/*
```

In the example above DD NAME is used as follows:

DD Name	Description
MDHPTRCE	The data set for trace output. You can change the DDNAME by altering the value for -t at the command line.
MDHPPRNT	The data set for standard output.
MDHPDIMV	The file that is used for the Dimensions environment variables. The name must match the name specified in the _CEE_ENVFILE control.
MDHPPARM	The input file used to extend the 'PARM=' statement. The portion after the forward slash '/' in the 'PARM=' statement is processed first, followed by the controls in columns 1-72 of the stream pointed to by MDHPPARM.
MDHPBRD	An optional name that you specify in the -b switch. Can be a USS file or a PDS member.

The Secondary Build Execution Monitor

The Secondary Build Execution Monitor (SBEM) is an MVS only generalized batch execution utility. The SBEM reads and executes JCL-like syntax; it also processes comments of the form ///*SBEM to control monitoring and error returns. You can run the SBEM in a JCL stream of its own using the MDHBSBM0 template, or from the USS command line. Running the SBEM from USS is useful when you want to execute JCL synchronously, for example, to populate a build area.

The current implementation of the SBEM is suitable for builds. The SBEM parses the whole JCL stream before initiating any work. It then executes the stream step by step, wrapping each executed program with calls to the SVC monitoring facility. At the end point(s) it replies to the PBEM with a task status.

There is no limit to the number of statements that you can code.

For details about using SBEM directives in build templates see the chapter *Writing Build Templates* in the *Developer's Reference*.

SBEM Parameters

Parameter	Default value	Purpose
-s		Traces storage.
-n		Returns information messages to the server.
-m		Traces message processing.
-o <filespec>	DD:MDHSPRNT	Specifies where to deliver standard output. Use \$ for stdout.
-I <filespec>	DD:MDHSIN	Specifies the location of the input file. Use \$ for stdin.
-t <filespec>	DD:MDHSTRCE	Specifies where to direct trace outputs.
-a <address>		Specifies the address (TCP/IP or DNS name) of the PBEM.
-p <port number>		Specifies the port for the PBEM connection.
-b <project name>		Specifies the name of the build project.
-e		ABENDs if allocation fails.
-f		ABENDs in result processing.
-y <n>		Specifies the allocation number after which the SBEM fails (the allocator returns NULL).
-r		If processing fails, returns a non-zero return code.
-w <instream file mask>	DD:MDHSWORK	Specifies a library to be used for instream data streams. This specification can include '%d' to identify the actual member to use. The SBEM generates a different name for each instream member.
Note:		
If you are using the SBEM from USS, the work data set must be of the form:		
-w "''//<data set name>(S%05d)'"		
where <data set name> is either a pre-existent data set or is allocated by the SBEM prior to its first use.		
-v <SVC number>		Specifies the number to use for the Watcher SVC.
-z n		Specifies the deallocation number after which the SBEM fails.

Example JCL for Starting the SBEM

```

//STEP100 EXEC PGM=MDHLSBEM,DYNAMNBR=1500,
// PARM='POSIX(ON),TRAP(OFF),HEAPCHK(OFF),ENVAR("_CEE_ENVFILE=DD:DV")'
//STEPLIB DD DISP=SHR,DSN=MDH.V1010.MDHLLIB
//          DD DISP=SHR,DSN=MDH.V1010.MDHLLPA
//MDHSPARM DD *
-a MVSnode
-p 4059
-b "BUILD"
-t DD:MDHSTRCE
/*
/*      main execution summary goes in here
//MDHSPRNT DD SYSOUT=*
/*      only used if exit tracing is turned on
//MDHOUT DD SYSOUT=*
/*      trace from sBem
//MDHSTRCE DD SYSOUT=*
/*      Environment variables for Dimensions listener
//DV DD DISP=SHR,DSN=MDH.DIM671.PARM(MDHTDIMV)
/*      Common BOM library for this run
//MDHBOM DD SYSOUT=*
/*      SYSIN data gets copied here
//MDHWORK DD DISP=(NEW,CATLG,CATLG),
//          DSN=&&TEMP01,
//          UNIT=SYSDA,DSNTYPE=LIBRARY,
//          DCB=(LRECL=80,RECFM=FB,BLKSIZE=3120,DSORG=PO),
//          SPACE=(TRK,(20,20,0))
/*      BOM template comes from here
//MDHTMPLT DD DISP=SHR,DSN=MDHDEV.ISPF.TEMPLATE
/*      input JCL for execution - expanded script
//MDHSIN DD DATA,DLM='++'
<JCL stream>
++

```

The DD NAMES are typically used as follows:

DD NAME	Purpose
MSHSPARM	The extension of the 'PARM=' statement. Only columns 1-72 are used.
MDHSPRNT	The data set for the SBEM primary output; it lists the JCL being executed and identifies any statements that have an error. Also contains brief messages about the progress of the individual steps being executed.
MDHOUT	The data set for SVC debugging output. Only used if you specify the -t parameter.
MDHSTRCE	The data set for trace output. Use the -t parameter to specify another name.
MDHBOM	The data set where the bill of materials is produced. This bill of materials can be passed back to the build server to update relationships in the SCM configuration.
MDHTMPLT	Points at the template library that is used to format the bill of materials. The bill of materials is generated based on the MDHBOM0 template.

DD NAME	Purpose
MDHSIN	The data set for the SBEM JCL primary input. Can be one of the following: <ul style="list-style-type: none"> ■ instream ■ DATA with DLEM ■ from a member
MDHWORK	The container used to hold instream data until it is needed. Instream data must be in 80 byte records.



NOTE The SBem implements a subset of JCL constructs that cover most of the basic syntax. If a feature that you require is missing, submit an enhancement request via the Serena support web site.

Starting the SBEM from USS

You can also start the SBEM from a USS prompt, which can be useful when you use scripts. For example:

```
export DMWORKOPT=-w // 'MY.DATASET(S%%05d) '
#
# Use the common dimensions profile
#
. /serena/instance/.dmprofile
#
# - -r returns the rc so that the script has the execution rc returned
# - -o $ puts the sbem print to stdout
# - -w identifies a dataset pattern (e.g. 'FOO.A.CNTL(S%%05d)') for data
# storage
#
sbem -i $ -o $ -r $DMWORKOPT <<EOF
jcl goes here
EOF
export rc = $?
exit $rc
```

There is a similar template, `sbem_start.sh`, in the instance templates library that allows JCL to be sent as part of an REXEC and executed synchronously.

MDHLLNK0

Use the MDHLLNK0 utility before the link step to:

- Remove duplicate include statements from a link stream.
- Insert the footprint CSECT into a link deck.
- Use OPEN/CLOSE on members; you can use it to determine source dependencies that the link editor would see. The link editor uses DESERV, which the WATCH facility does not support.

Use MDHLLNK0 in the templates MDHBLNK0 and MDHBLNK1.

General Parameters

Parameter	Description
-t <filename>	Enables tracing to the file that you specify. You can use DD:FOO.
-z	Returns the code RC=0 even if some of the included files were not found.
-f	Includes the footprint CSECT.

Output Control Parameters

Parameter	Description
-u<n>	<p>Controls what action is taken when a linkage editor "NAME" control card is found in the input:</p> <ul style="list-style-type: none"> ■ -u0 No action ■ -u1 The output is opened only for reading. ■ -u2 (Default) The output is opened for reading, and if that fails, is opened for writing. This action allows the SBEM to detect the targets in all cases. However, you may not require this behavior in some builds.

Passthrough Control Parameters

Control cards can be read with one of the following options:

Option	Description
I	INCLUDE statements.
D	Dependency INCLUDE statements.
C	Control statements.
X	Executable OBJ code (or LOAD/NCAL) data.

These options control what is written to the output file MDHFLAT. The use of this output file is optional and you can pass the original input into the linker.

Option I- INCLUDES

INCLUDES for non-dependency files are always copied to MDHFLAT as they are essential for link processing. For example:

- INCLUDE OBJ(MODULE)
- INCLUDE NCAL(MODULE)
- INCLUDE LOAD(MODULE)

Option D- DEPENDENCY INCLUDES

INCLUDES for dependency files are not generally needed by the linker once all the real INCLUDES that they contain have been flattened out. However, you can request them as follows:

- -pd0: does not include dependency includes.
- -pd1: (default) includes dependency includes.

For example:

- INCLUDE DEP(MYDEPS)
- INCLUDE OBJ(MYDEPS)

The nature of the include is decided by the data that the member contains. A member that has no executable code, but does have other INCLUDE statements, is considered to be a dependency file.

Option C - Control Cards

Other control cards may be encountered and you can optionally include them in the output:

- -pc0: does not output control cards.
- -pc1: (default) outputs control cards

Example:

NAME FOOBAR(R)

Option X - Executable Code

There are three methods that you can use to handle modules that mix executable (OBJ) records with control statements:

- -px0: does not output the OBJ code, but continues scanning for embedded control cards (if the PDS is FB-80).
- -px1: outputs the OBJ data directly into MDHFLAT. This produces 'instream' OBJ records for the linker.
- -px2: (default) when an x record is encountered, stops scanning for includes, and marks this as a non-dependency file.

Order of Output Lines

You can optionally output the INCLUDE line before or after the data that the included file contains. If the control statements refer to particular modules, ordering is important, and -po1 (the default) is recommended. For example, some control statements refer to the CSECT that comes next in the data stream.

- -po0: uses the recursive order (depth first). The body will be expanded first.
- -po1: (default) uses the normal order. The INCLUDE line comes before the body.

Examples:

Content of the file MODA:

```
A-START
INCLUDE OBJ (MODB)
A-END
```

Content of the file MODB:

```
B-START
INCLUDE OBJ (MODC)
B-END
```

Content of the file MODC:

```
C-START
C-END
```

Content of MDHFLAT using -po0:

```
A-START
B-START
C-START
C-END
INCLUDE OBJ (MODC)
B-END
INCLUDE OBJ (MODB)
A-END
INCLUDE OBJ (MODA)
```

Content of MDHFLAT using -po1:

```
INCLUDE OBJ(MODA)
A-START
    INCLUDE OBJ(MODB)
B-START
    INCLUDE OBJ(MODC)
C-START
C-END
B-END
A-END
```

Example JCL

The following example JCL is similar to that generated by the templates MDHBLNK0 and MDHBLNK1.

```
//PRELINK EXEC PGM=MDHLLNK0,
// PARM='POSIX(OFF),TRAP(OFF) /-t DD:MDHTTRC %PRELKOPT. -u0'
//STEPLIB   DD  DISP=SHR,DSN=MDH.V1010.MDHLLIB
//           DD  DISP=SHR,DSN=MDH.V1010.MDHLLPA
//OBJECT    DD  DISP=SHR,DSN=LIB1.DEV.OBJ
//           DD  DISP=SHR,DSN=LIB1.UT.OBJ
//           DD  DISP=SHR,DSN=LIB1.ST.OBJ
//           DD  DISP=SHR,DSN=LIB1.REL.OBJ
//LNKLIB     DD  DISP=SHR,DSN=LIB1.DEV.LNKLIB
//           DD  DISP=SHR,DSN=LIB1.UT.LNKLIB
//           DD  DISP=SHR,DSN=LIB1.ST.LNKLIB
//           DD  DISP=SHR,DSN=LIB1.REL.LNKLIB
//SYSLIN     DD  DISP=SHR,DSN=LIB1.DEV.SYSLIN(LMODULE)
//DEPENDCY  DD  DISP=SHR,DSN=LIB1.DEV.DEPENDCY
//           DD  DISP=SHR,DSN=LIB1.UT.DEPENDCY
//           DD  DISP=SHR,DSN=LIB1.ST.DEPENDCY
//           DD  DISP=SHR,DSN=LIB1.REL.DEPENDCY
//IMPORT    DD  DISP=SHR,DSN=LIB1.DEV.IMPORT
//           DD  DISP=SHR,DSN=LIB1.UT.IMPORT
//           DD  DISP=SHR,DSN=LIB1.ST.IMPORT
//           DD  DISP=SHR,DSN=LIB1.REL.IMPORT
//           DD  DISP=SHR,DSN=CBC.SCLBSID
//           DD  DISP=SHR,DSN=CEE.SCEELIB
//SYSLMOD   DD  DISP=SHR,DSN=LIB1.DEV.LOAD(LMODULE)
//MDHFLAT   DD  DISP=SHR,DSN=LIB1.DEV.MDHFLAT(SMODULE)
//MDHLPRT   DD  SYSOUT=*
//MDHTTRC   DD  SYSOUT=*
```

The example JCL above uses the following DDNAMEs:

DD Name	Description
SYSLIN	The data set for the input stream.
MDHFLAT	The data set for the output stream for the link editor.
MDHLPRNT	The data set for messages.
MDHTTRC	The data set for trace. Use the -t parameter to change this name.
DDNAME	Other DDNAMEs are required if there are INCLUDE statements that reference them.

bldcomms

The bldcomms utility communicates from a build step back to the PBEM or a build server. You can run bldcomms from the Windows, UNIX, or USS command-line prompts, or from MVS batch.

To run bldcomms issue this command:

```
bldcomms monitormessage <node> <port> "<content of message>"
```

where:

Parameter	Description
<node>	Specifies the node number of the application expecting the message, typically the PBEM. In a template use the symbol %DMPBEMNODE.
<port>	Specifies a randomly generated port number used by the PBEM to communicate with its started tasks. In a template use the symbol %DMPBEMPORT.
"<content of message>"	<p>Messages have two possible formats:</p> <ul style="list-style-type: none"> ■ I=<step num> T=<timestamp> M=<qprefix messageq> ■ R=<num> I=<step num> T=<timestamp> <p>where:</p> <ul style="list-style-type: none"> ■ <step num> is the step number that identifies the BRD step that this message relates to (in templates use the symbol %DMSTEP.). ■ <timestamp> is the timestamp, in the format yyyyttddhhmmssuuuuuu, that is constructed by the PBEM as part of template processing and is checked to ensure a match. You can also use '*****' which always matches. ■ <qprefix messageq> is as follows: <ul style="list-style-type: none"> • 'q' is a quotation character. • 'prefix' is the prefix of a message. The last letter of the prefix message describes the severity of the message. E and W are always delivered to the build server, other levels (T, I, and D) are not. • 'message' is the content of the message. <p>For example:</p> <pre>"M='JMH001W This is a test message'"</pre> <ul style="list-style-type: none"> ■ <num> is the return code (an integer).

Using bldcomms on MVS

You can also use bldcomms on MVS, for example, with an Openmake build. The MVS version of bldcomms is MDHBLCM.

The following JCL example is for the first <content of message> format:

```
//STEP100 EXEC PGM=MDHBLCM,
// PARM=('POSIX(ON)',
//        '/monitormessage node port "R=&RC',
//        ' I=%DMSTEP. T=%TIMESTAMP."')
//STEPLIB DD DISP=SHR,DSN=MDH.V1010.MDHLLIB
//          DD DISP=SHR,DSN=MDH.V1010.MDHLLPA
//SYSPRINT DD SYSOUT=*
```

The following JCL example is for the second <content of message> format:

```
//STEP100 EXEC PGM=MDHBLCM,
// PARM=('POSIX(ON)',
//        '/monitormessage %DMPBEMNODE. %DMPBEMPORT. "I=%DMSTEP.',
//        ' T=%TIMESTAMP. M='''JMH001W This is a test message'''')
//STEPLIB DD DISP=SHR,DSN=MDH.V1010.MDHLLIB
//          DD DISP=SHR,DSN=MDH.V1010.MDHLLPA
```

In the example above the M= clause has a single quotation mark (') but as it is inside a PARM statement it must have a double quotation mark (""). Note also that the comma (,) after DMSTEP appears in the message to the PBEM but is ignored because the parameter is converted to an integer. So you must ensure that the comma does not occur somewhere where it is not handled.

There is no provision for very long messages.

Loading Multiple Members

To load multiple members into a Dimensions database use the JCL stream in the MDHLOAD data set. MDHLOAD is located in the following data set:

MDH.V1010.SUPP.EXAMPLES.CNTL

Instructions for customizing and using MDHLOAD are embedded in comments in the JCL, and the core is written in REXX.

Appendix D

Dimensions Build Security

Phase 1: REXEC	340
Phase 2: dmlibsrv	340
Phase 3: The PBEM	340
Phase 4: The SBEM	341
Phase 5: Output Collection	341
Dimensions and Build Area Accounts	341
The Default Userid	342
Dimensions Build Area Security	342
Using Build Areas Outside Dimensions Build	342

Phase 1: REXEC

A build starts when the Dimensions CM server 'dmappsrv' process sends an REXEC to a build node. You can also manually start a build using the REXEC command.

- If you execute the REXEC command manually, you must first AUTH to the build node, or use the /USER=<build area user> and /PASS=<build area password> options on the REXEC command.
- If Dimensions CM starts a build, the security credentials come from the AREA definition of the deployment area. These credentials are used in the same way as the REXEC options /USER and /PASS. If the area does not have credentials (which is an invalid setup for a deployment area), you may be prompted to enter the userid and password at build time. You can also issue an AUTH command prior to the BLD command.

Phase 2: dmlibsrsv

The first visible result of the REXEC and AUTH commands is a process called 'dmlibsrsv' on the remote machine. 'dmlibsrsv' runs with the operating system security credentials of the build area. If you are a UNIX superuser, you can see this process and check which user it is using, by using the 'ps -elf' command at the UNIX prompt. From SDSF on MVS you can see this process from the 'PS' display.

The process 'dmlibsrsv' now runs the build. All actions are performed from this user.

Phase 3: The PBEM

On UNIX, the PBEM (Primary Batch Execution Monitor) runs as a normal program (pbem.exe) using the current security credentials.

On MVS a batch job is submitted. The job runs under the security of the 'dmlibsrsv' process.



NOTE You can change the job card. For example, you can set up RACF surrogate authority, but this happens outside of Dimensions Build, and also applies to non-build jobs. The process is similar to the build user logging in to TSO/ISPF, using the editor on some JCL, and then typing SUBMIT. The job would probably run as the logged in user. However, it may run as someone else. For example you may have put USER=XXX in the job card. You may also have a JES exit, or a RACF rule that says jobs matching a certain standard (for example JOBNAME), should run as a particular user.

Jobcards

You can customize the jobcards that build steps and the controlling PBEM use by changing their templates. For example, for a normal build step, the jobcard is found at the top of the SBEM template, TEMPLATE(MDHBSBEM0). The jobcard for the PBEM is in TEMPLATE(MDHBPBM0). There are many template expansion variables available to allow

a flexible scheme for naming the jobs. For details of the template variables see *The Templating Language and Processor* chapter of the *Developer's Reference*



NOTE

- Changing these jobcards does not in itself change security in any way, unless the site has some additional RACF rules relating to jobnames and users.
- Care is needed to ensure that whatever jobname scheme is chosen, the PBEM and build steps can run together. For example, a build step cannot be given the same jobname as a PBEM controller, or a deadlock situation will result.

Phase 4: The SBEM

The SBEM (Secondary Build Execution Monitor) runs under the same authority as the PBEM batch job. The SBEM, as part of its build execution activities, attaches some programs, such as the COBOL compiler. The attached programs run under the same user as the SBEM.

Phase 5: Output Collection

When a build is complete, the related objects can be added to Dimensions CM. This process uses the build area credentials on the remote node, and the credentials of the user logged in to Dimensions CM (either the user logged into Dimensions Build, or the user logged in to a Dimensions client to issue a BLD command).

Scheduled Builds

When you set up a scheduled build you may need to specify the password for each user that requires access to the build area. This applies if you specified that a password is required at runtime when you attached a build area to a build configuration.

Dimensions and Build Area Accounts

For a remote node the Dimensions userid does not matter as the REXEC always supplies the build area credentials.



NOTE There are variables of the form DMXXXXX that have Dimensions information but these do not affect security. These variables exist to enable a template to create correctly named files and logs. The template writer can decide whether to use these variables, or use some other naming standard. For example, a mainframe template could create a dataset called "DMSYS.TEMP.XXX", where DMSYS is an uppercase version of the Dimensions user. However, this is just a name. The process will not be run as DMSYS, unless that is also the credential on the build area. For this to work, the area credentials will require access to create these data sets.



TIP The Dimensions userid and password are typically case sensitive. The RACF user and password are typically case insensitive. JCL dataset names have to be uppercase. To avoid problems, enter the MVS USER and PASSWORD in uppercase. However, if you are referring to the UNIX home directory, use lower case. (there is a template function to do this conversion)

The Default Userid

The Dimensions server initially tries to log in to a remote node using the same credentials as the server account. If you have a MVS RACF account called DMSYS with the same password as the Dimensions account DMSYS, the server will log in as this user. You can switch this behavior off. For information, see the description of the following variables in the *Dimensions for z/OS User's and Administrator's Guide*:

- DM_MVS_REJECT_USERS (in the MVS configuration file).
- DM_SKIP_SERVER_CRED_CHECK (in the server configuration file).

Dimensions Build Area Security

Because a build area definition includes security credentials, you can use Dimensions roles to restrict the use of areas to specific Dimensions users.

Using Build Areas Outside Dimensions Build

You can use build areas on their own outside of Dimensions Build.

When you use an area name directly, the directory that the area specifies is used automatically. For example, assume that MYAREA specifies the location MYNODE::USER.WORKAREA. The following commands achieve the same aim:

- FI <item> MYNODE::USER.WORKAREA.COBOL(PROG1)
- FI <item> MYAREA::COBOL(PROG1)

The method that uses MYAREA has two advantages:

- It is shorter.
- It logs on automatically using the credentials in the area.

Index

A

active item revisions 135
Ant and Dimensions Build
 archiving buildfile 294
 build areas 295
 Dimensions project required 295
 editing imported targets 297
 editing transition scripts 298
 importing buildfiles 295
 overview 294
 running imported build jobs 299
application rule templates
 about 179
 adding 180
 deleting 181
 examples 179
 modifying 180
 transition rule templates, adding 181
 transition rule templates, deleting 182
 viewing 180
architecture
 distributed platforms 321
 MVS platforms 319
auditing, build areas 138
automatic builds 136

B

bill of materials 275
bldcomms utility
 about 336
 parameters 336
 using, on MVS 337
BRD file 318, 324
build administration main window 140
build areas
 about 133
 active item revisions 135
 and deployment scripts 47
 auditing 138
 cannot be deleted if used in build configuration 197
 creating a deployment area 193
 creating a library cache area 195
 creating a work area 192
 deleting 197
 locations 133
 managed development areas 135
 modifying 197

 overview 191
 security and ownership 134
build configuration type options
 about 153
 adding 154
 deleting 156
 modifying 155
build configuration types
 about 152
 adding 154
 deleting 156
 modifying 155
 Openmake 153
 viewing 154
build configurations
 about 184
 basic operations 186
 copying 189
 creating 186
 deleting 190
 modifying 188
 properties 190
 tips 225
 viewing 186
build environment for MVS, setting up 150
build flow 129
build job schedules
 about 250
 adding 252
 deleting 259
 frequency 255
 modifying 256
 range of recurrence 255
 scheduler service, configuring 250
 start time, specifying 254
 viewing 252
build jobs
 cancelling 269
 execution on mainframe 263
 rebuilding 272
build management tab, build administration 141
build monitor event errors, viewing 314
build monitoring tab, build administration 145
build notification events
 about 279
 adding 286
 deleting 288
 editing 287
 viewing 285
build notification subscriptions
 about 280

- adding 288
 - deleting 290
 - editing 289
 - viewing 285
 - build notification templates**
 - about 278
 - adding 282
 - copying 284
 - deleting 284
 - editing 283
 - viewing 281
 - build notifications**
 - about 278
 - e-mails, format of 278
 - example 279
 - notification service, configuring 280
 - variables, in templates 278
 - build option groups**
 - about 163
 - adding 165
 - build options, adding 165
 - build options, deleting 168
 - build options, modifying 167
 - deleting 167
 - examples 163
 - modifying 166
 - viewing 164
 - build options**
 - about 219
 - creating 219
 - deleting 221
 - modifying 221
 - overview
 - build outputs, capturing** 136
 - build privileges** 136
 - build projects, see Dimensions projects**
 - Build Request Definition file** 324
 - build roles** 136
 - build scheduling tab, build administration** 143
 - build security** 339
 - build settings** 151
 - build stage, see deployment stage**
 - build targets**
 - Build Plan page of Create New Target Wizard
 - 204
 - creating 202
 - deleting 205
 - modifying 205
 - Options page of Create New Target Wizard
 - 204
 - overview
 - Scripts page of Create New Target Wizard
 - 203
 - Target page of Create New Target Wizard 203
 - build terminology** 127
 - build tool options**
 - adding 159
 - deleting 162
 - modifying 161
 - build tools**
 - about 157
 - adding 159
 - deleting 162
 - example 158
 - external XML file format 157
 - importing, from XML file 160
 - modifying 161
 - viewing 158
 - building, automatically** 136
- C**
- capturing, build outputs** 136
 - change document, see request**
 - closed-loop builds** 136
 - contacting technical support** 16
- D**
- debugging**
 - BRD file 318
 - locating temporary files 317
 - preserving temporary files 317
 - temporary directory hidden 318
 - temporary files 317
 - dependencies** 273
 - requires preserving targets 273
 - deployment area** 47, 193
 - deployment scripts**
 - and build areas 47
 - deployment stage** 14
 - Dimensions CM**
 - documentation set 14
 - product components 13
 - Dimensions projects**
 - Disassembler quick start tutorial** 65
 - documentation set** 13, 14
- E**
- error messages**
 - Area is in use 315
 - Attempt to close invalid connection 316
 - Failed to authenticate to the build agent 316
 - Failed to find product-specific upload rules 316
 - No build areas found for the selected configuration 316
 - Template pbem_stop.cmd open failed 316

You do not have a role to extract item 316
 errors, build monitor events 314
 exporting
 overview 222
 to XML file 223
 external item revisions 137

F

filter conditions
 editing 242
 removing all conditions 242
 removing specific conditions 242
 filtering
 introduction 236
 build jobs, by date and time 238
 build jobs, by user name 236
 Dimensions projects 240
 filter conditions, editing 241

G

global script 215

I

impact analysis data (Openmake), producing 309
 importing
 from Ant build.xml 224
 from XML file 223
 overview 222
 inbox 14
 inputs
 adding new source as input 206
 adding new target as input 206
 overview 206
 introduction, to Dimensions Build 126

L

library cache area 195

M

managed development areas 135
 MDHJLOAD, loading multiple members 337
 MDHLLNK0 utility
 about 331
 example JCL 334
 monitoring builds
 immediately after launch 262
 past builds 270

running builds 268
 MVS
 recursive search not available 201
 wildcard patterns not available 198
 MVS build environment, setting up 150

N

notification tab, build administration 146
 notifications, see build notifications

O

Openmake, integrating with Dimensions Build
 build configuration type 304
 concatenation, in search paths 305
 impact analysis data, producing 309
 integration steps 303
 search paths 304
 TGT files, importing to a build configuration 307
 options, see build options

P

PBEM, see Primary Batch Execution Monitor
 pending list, see inbox
 placeholder item revisions 138
 preservation policies
 about 137
 external item revisions 137
 placeholder item revisions 138
 preserving targets 137
 Primary Batch Execution Monitor
 about 324
 example JCL 327
 parameters 325
 running 324
 printing manuals 16
 product name changes 14
 project 14
 project working location 14

Q

quick start tutorials
 C Quick Start Part One 19
 C Quick Start Part Two 49
 MVS 65

R

recursive search 201
request 14
rewrite
 compared to rollback 244
 example 246

rollback
 compared to rewrite 244
 example 244

running builds
 all builds 230
 error related to build areas 231
 from Build Scheduling tab 231
 Launch New Build 231
 previous versions 232
 single builds 228

S

SBEM, see Secondary Batch Execution Monitor

scenarios, usage 131

scheduler service, build administration 250

scheduling builds 250

Scripts

 overview 210

scripts

 deleting build configuration script 218
 deleting transition script 218
 modifying build configuration script 216
 modifying transition script 217
 using a Dimensions-controlled file 214
 using a file in the build area 214
 using global script 215
 viewing build configuration script 215
 viewing transition script 216

search paths, setting up 135

Secondary Batch Execution Monitor

 about 327
 example JCL 329
 parameters 328
 starting, from USS 330

security 339

security and ownership, of build areas 134

Serena ChangeMan Builder, see Openmake, integrating with Dimensions Build

show progress details 263

sources

 adding from file 198
 adding individual files 200
 and wildcard patterns 198
 deleting 202
 modifying 201
 overview 198
 recursive search 201

wildcards not available on MVS 198
supported platforms 126

T

targets, *see* build targets

technical support

 contacting 16

temporary files 317

terminology changes

 build area 14
 build stage 14
 change document 14
 custom list 14
 deployment area 14
 deployment stage 14
 in box 14
 library cache area 14
 list of 14
 pending list 14
 privilege 14
 project 14
 project working location 14
 request 14
 request list 14
 user interface profile 14
 user list 14
 work area 14
 workset 14
 workset root directory 14

terminology, Dimensions Build 127

TGT files, importing to a build configuration 307

transition rule template build option groups

 adding 178
 deleting 179

transition rule template build options

 adding 176
 deleting 178
 modifying 177

transition rule template input masks

 adding 173
 deleting 174
 modifying 173

transition rule template output masks

 adding 174
 deleting 175
 modifying 175

transition rule templates

 about 169
 adding 170
 deleting 172
 examples 169
 modifying 171
 script types, selecting 171
 viewing 170

troubleshooting 313
 build monitor event errors, viewing 314
 error messages 315
tutorials
 C Quick Start Part One 19
 C Quick Start Part Two 49
 MVS 65

U

user list, *see* custom list and request list
user scenarios 131

V

versions
 allowed operations 244
 delete not allowed 247
 possible operations 222

W

work area 192
workset root directory, *see* project working location
workset, *see* project

Z

z/OS mainframe nodes, creating definitions for
 150

