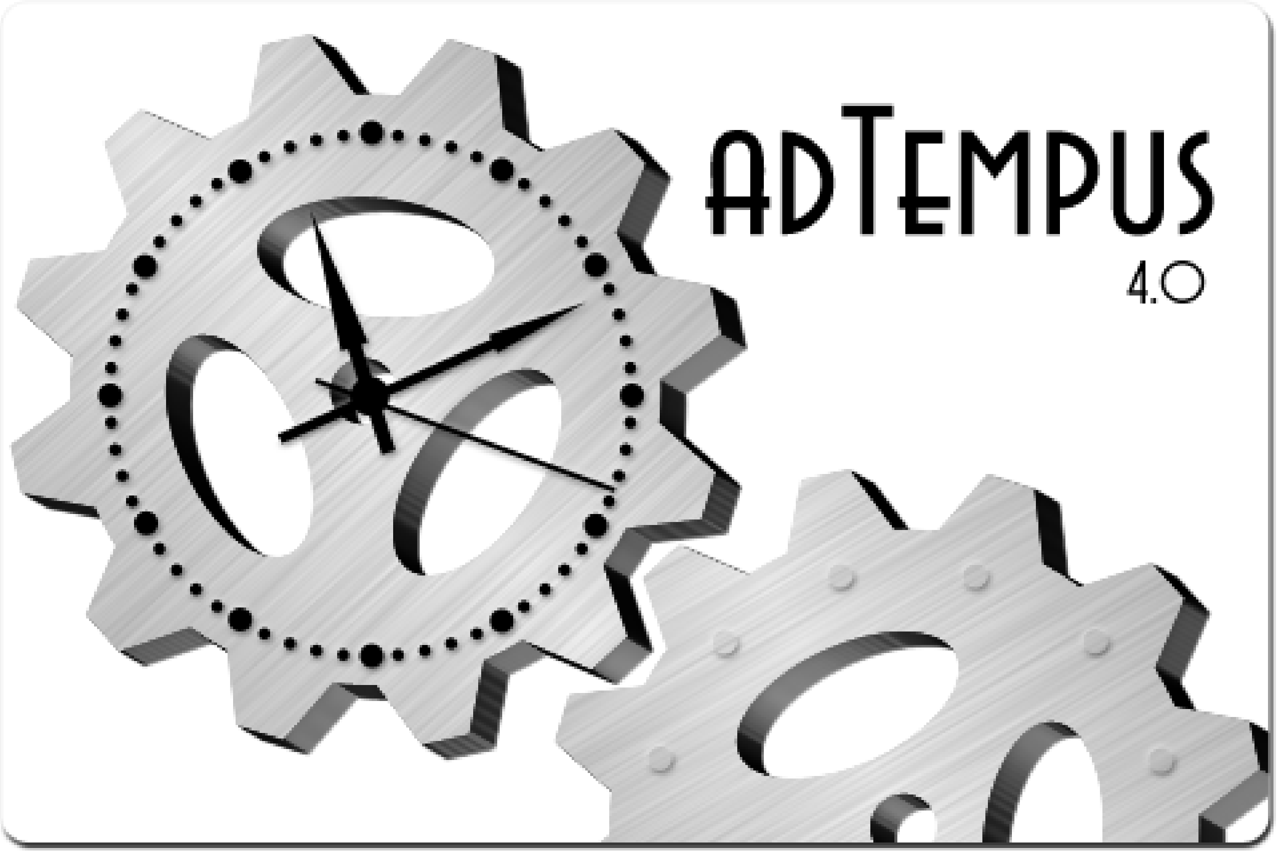
adTempus 4 User Guide



adTempus 4.5

Generated 12/19/2017

Contents

adTempus 4 User Guide **1**

Welcome to adTempus **68**

About adTempus Help 68

Getting Started 68

Additional Help Resources **69**

What's New in adTempus 4 ..... **70**

Installation and Configuration 70

Security 70

Job Variables 70

Jobs 71

Job Groups 71

Job Queues 72

Tasks 72

Program Execution Task 72

Database Operation Task 72

E-Mail Processing Task 72

File Transfer Task 72

File Compression/Uncompression 72

Job Variable Update Task 73

Computer Restart Task 73

Service Control Task 73

Notification Task 73

Conditions 73

Job Condition 73

Job Variable Condition 74

Triggers 74

E-Mail Trigger 74

File Trigger 74

Job Trigger 74

Schedule Trigger 75

Scripting 75

Auditing, Change Tracking, and Snapshots 75

E-Mail Notification for Alerts 76

Job Monitor View 76

Distributed Scheduling Changes 76

Other Console Improvements 76

Export/Import 77

Job History 77

Server Trust Relationships 78

New Reporting Platform 78

Miscellaneous Changes 78

Contacting Us 80

Sales 80

Technical Support 80

Product Feedback 80

Introduction 81

Overview 81

Key Concepts 81

Introduction to Key Concepts 81

Jobs, Steps, and Tasks 82

Triggers 83

Conditions 83

Responses, Events, and Actions 83

Job Groups 83

Job Queues 84

Multiple Steps vs Separate Jobs 84

Comparison of Triggers and Conditions 87

Job Variables 88

Job Variable Inheritance 88

Using Job Variables 89

Variable Formatting 89

Resources 90

Cycle ID 90

How the Cycle ID is Assigned to a Job 90

Cycle Scopes 90

How to Update the Cycle ID 91

Viewing the Cycle ID for an Instance 91

Credential Profiles 91

Features and Benefits 91

Simplified Password Management 92

Delegated Account Use for Improved Security 92

Simplified Data Entry 92

Creating and Using Credential Profiles 92

Managing User Profiles 92

Changing Credentials 92

Notification Recipients 93

Using Scripts in adTempus 93

Kinds of Scripts 94

Task Scripts 94

Extension Scripts 94

Shared Scripts 95

Script Libraries 95

Installation 97

Installing 97

adTempus Components 97

Installation Process 97

Software Installation 98

Data Collection 98

Setup Type 98

Engine Mode 98

Installation Wizard Completed 99

License Configuration 99

Database Installation and Configuration 99

Engine Mode 100

Standalone 100

Master 100

Agent + Standalone 100

Agent Only 100

System Requirements and Prerequisites 101

101

System Requirements 101

Disk Space Requirements 101

Prerequisites 101

Database Requirements 101

Installing on Windows Server 2003 101

Use the Correct Setup Package 102

Windows Installer 4.5 Required 102

Additional Prerequisites 102

SQL Server 2008 Express Used Instead of SQL Server 2012 102

Upgrading from Previous Versions 102

Side-by-Side Installation 103

Database Changes 103

adTempus Service Account Requirements 103

Upgrading Multiple Servers 103

Linked Servers 103

Security Changes 104

Additional Breaking Changes 104

adTempus Console does not work across major versions 104

The adTempus export/import format has changed 104

WMI trigger selection script changes 104

Job Log message codes for Scripts 104

No more override of held jobs 104

User interaction setting changes 105

The adTempus API has changed 105

New Web client 105

Upgrading from adTempus 2.x 105

Distributed Scheduling Configuration 105

Security Changes from Version 2 105

Side-by-Side Installation 106

Installation Process 106

Side-by-Side Installation with Version 2 or 3 106

Licensing 107

Licensing Overview 107

License Requirements 107

Distributed Scheduling Agent Licenses 107

Evaluation Mode 107

Entering License Information 108

License Agreement 108

Software Product License 108

Disclaimer of Warranty 112

Database Installation and Configuration 113

Database Installation and Configuration 113

Overview 113

Configuration Process for New Installation 113

Configuration Process for Upgrade 113

Database Selection 113

Using the Default Database Engine 114

Using Your Own Database Server 114

SQL Server Express 115

Using a Standalone SQL Server Instance 115

Installation Requirements 116

Security Setup 116

Hosting Multiple adTempus Instances 116

Database Backups 116

Enabling SQL Server Security 117

Database Configuration Wizard Reference 119

SQL Server Express Download 119

MSDE Detected 119

Database Upgrade 119

SQL Server Express Upgrade 120

SQL Server Location 120

SQL Server Location 120

Use a named instance of SQL Server 120

Standalone SQL Server 120

SQL Server Authentication 121

Database Name 122

Database Already Configured 122

Database Credentials 122
Database Credentials 122
Windows Authentication 122
SQL Server Authentication 123
SQL Server Computer Account 123
Select Actions 123
Administrator Provisioning 124
Configuration Progress 124
Successful Completion 124
Failure 124
Uninstalling 125
License Deactivation 125
Database Removal 125
Multiple adTempus Server Instances 126
What is an Instance? 126
Requirements for Multiple Instances 126
Managing Instances 127
Creating Additional Instances 127
Removing Instances 127
Connecting to Instances 127
Software Updates 127
Console-Only Installation 128
Getting Started 129
Getting Started 129
First-Time Setup 129
Security Settings 129
Notification 129
Holidays 130

Shared Schedules 130

How To: Create a Basic Job 130

Next Steps 131

adTempus Objects 133

Job 133

Job 133

Security 133

Job Properties 133

Job General Page 134

Name 134

Job ID 134

Queue 135

Priority 135

User Account 135

Run with highest privileges 135

User Interaction 135

Hold Status 136

History Retention 136

Tags 136

Documentation Page 137

Description/Notes 137

Variables Page 137

Job Recovery Page 137

Run on startup if last run missed 137

Restart 137

Run on startup if adTempus was shut down while the job was running 138

Run on startup if adTempus terminated unexpectedly while the job was running 138

Restart From 138

Pass most recent checkpoint to job 138
Triggers Page 138
Triggers 139
Multiple Instances 139
Update the Cycle ID for this job's scope when the job runs 139
Conditions Page 140
Condition Criteria 140
If Conditions are not satisfied 140
Conditions List 140
Steps Page 140
Step Execution Sequence 141
Steps 141
Job Resources Page 141
Resource Failure 141
Resources List 142
Job Responses Page 142
Job Security Page 144
Steps Overview 145
Checkpoints 145
Viewing Checkpoints 146
Example: Using Checkpoints in a Batch File 146
Tasks 147
Tasks 147
Basic Tasks 147
Database Tasks 148
E-Mail and Notification Tasks 148
File Tasks 148
Network Tasks 148

Additional Tasks 148
Computer Shutdown Task 148
Computer Shutdown Task 148
Computer Shutdown Task Properties 149
Property Pages 149
General 149
Name for this step (optional) 149
Enable this step 149
Description/Notes 149
Conditions 149
Condition Criteria 149
If Conditions are not satisfied 150
Conditions List 150
Variables 150
Responses 150
Restart Options 151
Target Computer 151
After displaying warning message, wait __ seconds before shutting down 151
Restart the computer 151
Force all applications to close 151
Display the following additional information in the warning message 151
Database Operation Task 152
Database Operation Task 152
Reviewing Task Output 152
Database Operation Task Properties 152
Property Pages 153
General 153
Name for this step (optional) 153

Enable this step 153

Description/Notes 153

Conditions 153

Condition Criteria 153

If Conditions are not satisfied 153

Conditions List 153

Variables 153

Responses 154

Database Connection 154

Database Type 154

Specify custom connection string 155

Database Connection 155

Database Authentication 155

Test Connection 155

Database Operation 155

Execute SQL 155

Select a scalar value into a Job Variable 156

Select data into a DataSet 156

Execute database job 156

E-Mail Processing Task 157

E-Mail Processing Task 157

E-Mail Processing Task Properties 157

Property Pages 157

General 157

Name for this step (optional) 157

Enable this step 157

Description/Notes 158

Conditions 158

Condition Criteria 158

If Conditions are not satisfied 158

Conditions List 158

Variables 158

Responses 158

Message Source 159

Mail connection to use 159

IMAP Folder 159

Delete selected messages 159

Clear list of previously-processed messages 160

Message Selection 160

Selection Rules 160

Selection Script 160

Message Saving 160

Save format 160

Save messages to folder 161

Job Variables 161

File Operation Tasks 161

File Transfer Task 162

File Transfer Task 162

File Transfer Task Properties 162

Property Pages 162

General 162

Name for this step (optional) 162

Enable this step 163

Description/Notes 163

Conditions 163

Condition Criteria 163

If Conditions are not satisfied 163

Conditions List 163

Variables 163

Responses 163

Source 164

Action 164

Location 164

Source Root 164

Specify all files/paths relative to a single directory 164

Include all subdirectories 165

Specify absolute paths 165

Include Files 165

Exclude Files 165

Destination 166

Location 166

Destination Directory 166

Preserve source directory structure 166

File Overwrite 166

Make backups of existing files before replacing 167

Job Variables 167

File Compression Task 167

File Compression Task 167

File Compression Task Properties 168

Property Pages 168

General 168

Name for this step (optional) 168

Enable this step 168

Description/Notes 168

Conditions 168

Condition Criteria 168

If Conditions are not satisfied 168

Conditions List 168

Variables 169

Responses 169

Source 169

Location 169

Specify all files/paths relative to a single directory 170

Include all subdirectories 170

Specify absolute paths 170

Include Files 170

Exclude Files 171

Delete files after they are added to the archive 171

Destination 171

Compression Type 171

Location 171

Compress To 171

Store path information in archive 171

Existing Archive 171

Make a backup of the existing file before replacing or updating it 172

Encryption 172

Job Variables 172

File Uncompression Task 172

File Uncompression Task 172

File Uncompression Task Properties 173

Property Pages 173

General 173

Name for this step (optional) 173

Enable this step 173

Description/Notes 173

Conditions 173

Condition Criteria 173

If Conditions are not satisfied 174

Conditions List 174

Variables 174

Responses 174

Source 175

Location 175

Files to uncompress 175

Compression Type 175

Delete archives after files are extracted 175

Encryption 175

Destination 175

Location 175

Uncompress To 176

Create a subdirectory for each archive 176

Use path names from archive 176

File Overwrite 176

Make backups of existing files before replacing 176

Job Variables 176

File Selection Wildcards 177

Job Variable Update Task 178

Job Variable Update Task 178

Job Variable Update Task Properties 178

Property Pages 178

General 178

Name for this step (optional) 178

Enable this step 178

Description/Notes 179

Conditions 179

Condition Criteria 179

If Conditions are not satisfied 179

Conditions List 179

Variables 179

Responses 179

Variable Update 180

Variable to Update 180

Set value to 180

Increase/decrease value by 180

Scope 180

Notification Task 181

Notification Task 181

Notification Task Properties 182

Property Pages 182

General 182

Name for this step (optional) 182

Enable this step 182

Description/Notes 182

Conditions 182

Condition Criteria 182

If Conditions are not satisfied 183

Conditions List 183

Variables 183

Responses 183

Notification Message 184

Subject of message 184

Message to send 184

Notification Severity 184

E-Mail Sender 184

Recipients 184

Recipients Defined in adTempus 184

Additional Recipients 184

Attachments 184

Save these files in the job history 184

Include captured console output 185

Process Termination Task 185

Process Termination Task 185

Process Termination Task Properties 185

Property Pages 185

General 186

Name for this step (optional) 186

Enable this step 186

Description/Notes 186

Conditions 186

Condition Criteria 186

If Conditions are not satisfied 186

Conditions List 186

Variables 186

Responses 186

Process Termination 187

Process to terminate 187

Also terminate any child processes of the target process 187

Job Variables 188

Program Execution Task 188

Program Execution Task 188

Program Execution Task Properties 189

Property Pages 189

General 189

Name for this step (optional) 189

Enable this step 189

Description/Notes 189

Conditions 189

Condition Criteria 189

If Conditions are not satisfied 190

Conditions List 190

Variables 190

Responses 190

Execution Target 191

Target Type 191

Target 192

Command-Line Parameters 192

Use a script to specify command-line parameters 192

Skip this step if the process is already running 192

Common Settings 193

Startup Directory 193

Capture screen output from console-mode program 193

Execution Priority 193

Advanced 193

Limit execution to __ seconds 193

Startup Determination 194

Success Criteria 194

Calling Shared Batch Files 195

Sharing a Batch File 195

New Batch File 195

Sharing an Existing Batch File 196

Changing Shared Batch File References 196

Process Termination Task 196

Process Termination Task 196

Process Termination Task Properties 197

Property Pages 197

General 197

Name for this step (optional) 197

Enable this step 197

Description/Notes 197

Conditions 197

Condition Criteria 197

If Conditions are not satisfied 197

Conditions List 197

Variables 198

Responses 198

Process Termination 198

Process to terminate 198

Also terminate any child processes of the target process 199

Job Variables 199

Script Execution Task 199

Script Execution Task 199

Script Execution Task Properties 200

Property Pages 200

General 200

Name for this step (optional) 200

Enable this step 201

Description/Notes 201

Conditions 201

Condition Criteria 201

If Conditions are not satisfied 201

Conditions List 201

Variables 201

Responses 201

Script 202

Script to Execute 202

Execute a script stored in adTempus 203

Execute a script stored in an external file 203

Run isolated from other scripts 203

PowerShell Options 203

Capture console output from script 203

Command-Line Parameters 203

Options 203

Success Criteria 203

Time limit 204

Service Control Task 204

Service Control Task 204

Using adTempus to Automatically Restart a Failed Service 204

Service Control Task Properties 205

Property Pages 205

General 205

Name for this step (optional) 205

Enable this step 205

Description/Notes 205

Conditions 205

Condition Criteria 205

If Conditions are not satisfied 206

Conditions List 206

Variables 206

Responses 206

Service Control 207

Service Location 207

Service 207

Control Type 207

Options 208

Monitor the service 208

Stop the service if the job is terminated 208

Web Request Task 208

Web Request Task 208

Web Request Task Properties Window 209

Property Pages 209

General 209

Name for this step (optional) 209

Enable this step 209

Description/Notes 209

Conditions 210

Condition Criteria 210

If Conditions are not satisfied 210

Conditions List 210

Variables 210

Responses 210

Web Request 211

URL to Request 211

Credentials 211

Save the page or file returned by the server 211

Job Variables 212

Triggers 212

Triggers 212

Available Triggers 212

Computer Monitor Trigger 213

Computer Monitor Trigger 213

Computer Monitor Properties 214

Property Pages 214

General 214

Name for this trigger (optional) 214

Enabled 214

Minimum Interval 214

Description/Notes 214

Job Variables 214

Target 214

Target 214

Test Method 214

Response Evaluation 215

Use regular expressions 215

Test Interval 215

Trigger at test interval until successful response received 216

Trigger again when successful response received 216

Additional Options 216

Identity 216

Job Variables 217

Using to Monitor Your Web Application 218

A Simple Test 218

Doing More 218

Using the Results 219

Event Log Monitor Trigger 219

Event Log Monitor Trigger 219

Event Log Monitor Properties 220

Property Pages 220

General 220

Name for this trigger (optional) 220

Enabled 220

Minimum Interval 220

Description/Notes 220

Job Variables 220

Event Selection 220

Log 220

Sources 221

Category 221

Types 221

Include Event IDs 221

Exclude Event IDs 221

Select events whose message matches 221

Use regular expressions 221

Use a script to select events 222

Job Variables 222

E-Mail Trigger 222

E-Mail Trigger 222

E-Mail Trigger Properties 223

Property Pages 223

General 223

Name for this trigger (optional) 223

Enabled 223

Minimum Interval 224

Description/Notes 224

Job Variables 224

Message Source 224

Mail connection to use 224

IMAP Folder 224

Delete selected messages 224

Clear list of previously-processed messages 224

Check for new messages every 225

Selection Criteria 225

Selection Rules 225

Selection Script 225

Trigger separately for each message 225

Message Saving 225

Save format 225

Save messages to folder 226

Job Variables 226

E-Mail Selection Filter 227

Settings 227

Filter based on 227

Filter on property 227

Matching rule 227

Value to match 228

Use Regular Expressions 228

Examples 228

File Trigger 228

File Trigger 228

Changes Made While adTempus is Not Running 229

Removable Drives and Network Drives 229

File Trigger Properties 230

Property Pages 230

General 230

Name for this trigger (optional) 230

Enabled 230

Minimum Interval 230

Description/Notes 230

Job Variables 230

File Selection 230

File Location 230

Files 231

Polling Interval 231

Trigger Conditions 231

Trigger Mode 232

Trigger a single instance as soon as any matching file is found 232

Include all matching files in FileName variable 232

Trigger a separate instance for each matching file 233

Trigger a single instance once all file specifications have been met 233

Minimum Time Between Triggers 233

Additional Options 234

Wait for exclusive access to the file 234

Wait until __ seconds after the file's last modification 234

Capture the file 234

Selection Script 234

Job Variables 234

File Specification Properties 235

File Specification 235

Include subdirectories 235

Job Trigger 236

Job Trigger 236

Job Trigger Properties 236

Property Pages 236

General 236

Name for this trigger (optional) 236

Enabled 237

Minimum Interval 237

Description/Notes 237

Job Variables 237

Job Trigger 237

Job Trigger Target Rule Properties 237

Settings 238

Enable this rule 238

Trigger based on job 238

Rule 238

Instance 238

Ignore manual (on-demand) instances of job 239

Process Trigger 240

Process Trigger 240

Process Trigger Properties 240

Property Pages 240

General 240

Name for this trigger (optional) 240

Enabled 241

Minimum Interval 241

Description/Notes 241

Job Variables 241

Process Criteria 241

Trigger Conditions 241

Use a script to select processes 241

Description/Notes 242

Job Variables 242

Schedule Trigger 242

Schedule Trigger 242

Multiple Schedules 242

Schedule Trigger Properties 243

Property Pages 243

General 243

Name for this trigger (optional) 243

Enabled 243

Minimum Interval 243

Description/Notes 243

Job Variables 244

Schedules 244

Options 244

System Clock Changes 244

Rerun if clock is set back 244

Run once if clock is set forward so that one or more scheduled executions are missed 244
Time Zone 245
Local time zone on the computer where the job is triggered 245
A specific time zone 245
The time zone of the Master with which the Agent is associated 245
Holidays 245
Use the following set of holidays 245
When the job is scheduled to run on a holiday 246
Execution Schedule 246
Execution Schedule Properties 247
Property Pages 247
Date Selection 247
Enable this schedule 247
Optional Descriptive Name 247
Use this shared schedule 247
Active Range 248
Sharing 248
Make this schedule available for use by other jobs 248
Name for shared schedule 248
Time Selection 248
Trigger every... 248
Trigger at these times 248
Randomize Start Time 248
Date Rule Properties 249
Optional name for this selection 249
Enabled 249
Months 249

Trigger On 249

Selected days of month 249

Selected days of week 250

Days specified using a floating rule 250

Days specified relative to end of month 251

Days specified relative to Easter 251

Selecting Days of the Week 251

Selected days of week 251

Days specified using a floating rule 251

Matching Weekdays, Business Days, Weekends, Holidays for Floating Day Rules 252

Holidays 253

Startup Trigger 253

Startup Trigger 253

Avoiding Duplicate Processes 253

Suppressing Startup Jobs 254

Startup Trigger Properties 255

Property Pages 255

General 255

Name for this trigger (optional) 255

Enabled 255

Minimum Interval 255

Description/Notes 255

Job Variables 255

Startup Trigger 255

Wait ___ minutes before starting job 255

WMI Trigger 256

WMI Trigger 256

Changes from Previous Versions 256

WMI Trigger Properties 256

Property Pages 257

General 257

Name for this trigger (optional) 257

Enabled 257

Minimum Interval 257

Description/Notes 257

Job Variables 257

WMI Events 257

Namespace 257

Query 257

Use a script to select events 258

Conditions 258

Conditions 258

How Conditions Work 258

Responding to Unmet Conditions 259

Adding and Editing Conditions 259

Ignoring Conditions 259

Viewing the Status of Conditions 259

File Condition 260

File Condition 260

File Condition Properties 260

Property Pages 260

General 260

Name for this condition (optional) 260

Enabled 261

Description/Notes 261

Condition Wait 261

Condition Wait 261
Condition Polling 261
Once satisfied, do not re-evaluate when evaluating other conditions for this instance 261
File Condition 262
File Specification 262
Criterion 262
Include subdirectories 263
Only consider files modified since the last execution of the job 263
Wait for exclusive access to the file 263
Job Condition 263
Job Condition 263
Job Condition Properties 264
Property Pages 264
General 264
Name for this condition (optional) 264
Enabled 264
Description/Notes 264
Condition Wait 264
Condition Wait 264
Condition Polling 265
Once satisfied, do not re-evaluate when evaluating other conditions for this instance 265
Job Condition 266
Depend on Job 266
Rule 266
Instance 266
Ignore manual (on-demand) instances of job 267
Job Condition Instance Example 268

Job Variable Condition 269

Job Variable Condition 269

Job Variable Condition Properties 270

Property Pages 270

General 270

Name for this condition (optional) 270

Enabled 270

Description/Notes 270

Condition Wait 270

Condition Wait 270

Condition Polling 270

Once satisfied, do not re-evaluate when evaluating other conditions for this instance 271

Condition 271

Variable to test 271

Comparison rule 272

Compare to 272

Use Regular Expression 272

Process Condition 272

Process Condition 272

Process Condition Properties 273

Property Pages 273

General 273

Name for this condition (optional) 273

Enabled 273

Description/Notes 273

Condition Wait 273

Condition Wait 273

Condition Polling 274

Once satisfied, do not re-evaluate when evaluating other conditions for this instance 274

Process Condition 275

Process Name 275

Criterion 275

Script Condition 275

Script Condition 275

Script Implementation Guidelines 275

Waiting/Polling 275

State Persistence 276

Reporting Additional Status Information 276

Script Condition Properties 277

Property Pages 277

General 277

Name for this condition (optional) 277

Enabled 277

Description/Notes 277

Condition Wait 277

Condition Wait 277

Condition Polling 278

Once satisfied, do not re-evaluate when evaluating other conditions for this instance 278

Script Condition 279

Script to evaluate 279

Responses, Events, and Actions 279

Responses 279

Notes on using Responses 279

Response Properties 280

Events 280

Actions 280

Events 280

Response Event Properties 281

Use a script to determine whether the Response will be activated 281

Actions 281

File Capture Action 282

File Capture Action 282

File Capture Action Properties 283

Property Pages 283

Files 283

Files to Capture 283

Description/Notes 283

Send 283

E-mail captured files to the following recipients 283

Include captured console output 283

Subject of message 283

Message to send 283

Notification Severity 283

File Specification Properties 284

File Specification 284

Include subdirectories 284

Only capture files with the Archive attribute set 284

Only capture files modified since the job started 284

Delete files after they are captured 284

Job Control Action 284

Job Control Action 284

Job Control Action Properties 285

Property Pages 285

Action 285

Action to take 285

Wait __ seconds before restarting 286

Restart no more than __ times 286

Set the status of the calling job/step to "Resubmitted" 287

Applies To 287

Step 287

Run only the selected step 287

Responses 287

Checkpoint 288

Ignore conditions for the job 288

Ignore conditions for individual steps 288

Force a new instance of the job if necessary 288

Run only on same Agent 288

Run only on Master 288

Force job to run on Master 288

Variables 288

Job Variable Update Action 289

Job Variable Update Action 289

Job Variable Update Action Properties 289

Property Pages 290

General 290

Name for this action (optional) 290

Enabled 290

Description/Notes 290

Action 290

Variable to Update 290

Set value to 290

Increase/decrease value by 290

Scope 290

Notification Action 291

Notification Action 291

Notification Action Properties 292

Property Pages 292

Notification 292

Recipients 292

Subject of message 292

Message to send 292

Notification Severity 292

E-Mail Sender 293

Attachments 293

Script Action 293

Script Action 293

Script Action Properties 294

Optional name for this action 294

Script to Execute 294

Description/Notes 294

Response Example 294

Job Group 294

Job Group 294

Job Group Properties 295

Property Pages 295

Group 295

Name 295

Description 295

Hold Status 295

Cycle ID 296

Variables 296

Responses 296

Security 296

Job Queue 297

Job Queue 297

Limiting Job Execution 297

Holding and Releasing Queues 298

Distributed Scheduling 298

Job Queue Properties 298

Property Pages 299

Queue 299

Name 299

Hold Status 299

Resubmit pending jobs when adTempus restarts 299

Limit the number of jobs from this queue that can run concurrently 299

Description/Notes 300

Distributed Scheduling 300

Agent Mode 300

Run on Remote Agents 301

Evaluate Conditions On 301

Variables 301

Responses 301

Security 301

Remote Execution Options 302

Enable execution on this agent 302

Queue execution commands 302

Notification Recipients 303

Notification Recipient 303

Notification Recipient 303

Notification Recipient Properties 304

Property Pages 304

Recipient 304

Name 304

Enable this recipient 304

Severity 304

Addresses 305

Alerts 305

Security 305

Notification Address Properties 305

Property Pages 306

Address 306

Address Type and Address 306

Connection 306

Limit messages to 307

Schedule 307

Override schedule when severity is... 307

Schedule 308

Notification Group 308

Notification Group 309

Notification Group Properties 309

Property Pages 310

Group 310

Name 310

Enable this group 310

Severity 310
Members 310
Security 310
Notification Group Member Properties 311
Property Pages 311
Member 311
Recipient 311
Enable this member 311
Schedule 311
Override schedule when severity is... 311
Schedule 311
Scripts 312
Using Scripts in adTempus 312
Kinds of Scripts 312
Task Scripts 312
Extension Scripts 313
Shared Scripts 313
Script Libraries 313
Script 314
Script Implementation 314
.NET Scripts 314
Script Parameters 315
Windows Script Host Scripts 315
Script Parameters 315
Windows PowerShell Scripts 315
Script Properties Window 316
Property Pages 316
Script 316

Name 316

Shared 316

Description/Notes 316

Language 316

Included Script Libraries 317

Included Scripts 317

Included Batch Files 317

Referenced Assemblies 318

Script 318

Limit script execution 318

Run isolated from other scripts 318

Use 32-bit script host 319

Validate 319

Test 319

Security 319

Script Library 320

InlineFunctions Library 320

Script Library Code and Namespaces 320

WSH Scripting Languages 320

VB.NET 320

C# 321

Windows PowerShell 321

Script Library Properties 321

Property Pages 322

Script 322

Name 322

Description/Notes 322

Language 322

Referenced Assemblies 322

Included Script Libraries 322

Script 322

Validate 322

Security 322

Inline Functions 323

Defining Inline Functions 323

Calling Inline Functions 323

Inline Function Evaluation Language 324

Advanced Inline Scripting 325

Variable Treatment 325

Interacting with adTempus from Scripts 326

Integration Interface Members 326

Properties 326

Methods 327

Message Logging 327

Examples 328

Logging a Message 328

Getting and Setting Job Variables 328

Returning a Result From a Script 329

Script Security and Isolation 329

Remote Agent 330

Remote Agent 330

Remote Agent Properties 331

Property Pages 331

Agent 331

Agent Address 331

Instance 331

Port 331

Optional Name 331

Enable this agent 331

This computer is not in the same domain as the Master 331

Certificate Thumbprint 331

Advanced connection settings 331

Variables 332

Security 332

Security 333

Security Overview 333

Job Execution Security 333

adTempus Access Security 333

Security Changes from Prior Versions 334

Administrator Access 334

Security Logins 334

Security Groups 335

Additional Changes from Version 2 335

Security Inheritance 335

Security Editor 338

Users and Groups 338

Apply Permissions To 338

Permission 339

Inherit permissions from parent object(s) 339

Remove explicit permissions on all child objects 339

Security Login 339

Security Login Properties 340

General 340

Windows Authentication 340

adTempus Authentication 340
Enabled 340
Convert to standard Login 340
Description 340
Groups 341
Automatic Login through Group Membership 341
Converting to a Standard Login 342
Security Group 342
Security Configuration Guidelines 343
Security Configuration Guidelines 343
Permission Management Guidelines 343
User Groups 343
Object Groups 344
Object Creator 344
Initial Permission Setup 345
Creating Logins 345
Assigning Permissions 345
Default Permissions 345
Modifying Permissions 346
Permissions for All Objects 346
Permission to Create Jobs 346
Permission to Create Other Objects 347
Administrator Provisioning 347
Creating Administrators During Installation 347
Creating Administrators After Installation 348
Distributed Scheduling 349
Distributed Scheduling Overview 349
Distributed Scheduling Modes 349

Mirror 349

Slave 350

Load Balancing 350

Distributed Scheduling Installation 350

System Options on the Agent 351

Configuration 351

Distributed Scheduling Setup and Administration 351

Defining Agents 351

Configuring Queues 352

Configuring Jobs 352

Administering Distributed Jobs 352

Distributed Scheduling Recoverability 353

Communication 353

Queued Commands 353

Independent Operation 353

Managing Multiple Environments 354

Managing Multiple Environments 354

Managing Multiple Environments with Separate Instances 354

Environment Architecture 355

All environments on one computer 355

Each environment hosted on a separate computer 355

Installing adTempus Instances 355

Separate Computers 355

Same Computer 355

Considerations for Distributed Scheduling 356

Initial Configuration 356

Background: Kinds of Reference Data 356

Basic Settings 357

Notification Options 357

Linked Reference Data 357

Security 357

Propagating Initial Setup 358

Configuring Jobs for Multiple Environments 359

Use Job Variables 359

Use Notification Groups 360

Use the same Credential Profile, even if the user ID is different between environments 360

Migrating Jobs Between Environments 361

Selecting Objects for Import 361

Importing Security Settings 362

Auditing and Snapshots 364

Auditing and Snapshots Overview 364

Snapshot Overview 364

Restoring from a Snapshot 365

adTempus Console Reference 366

Console 366

View Reference 366

Welcome Page 366

Server Folders 367

Job Monitor 367

Show Active 367

Show Past 368

Show Next 368

Combined View 368

Filter 368

Failed Jobs 368

Failed Jobs 369

Troubleshooting Failed Jobs 370

Error Reported by adTempus 371

Errors Reported by Tasks 371

How adTempus Determines "Failure" 371

Diagnosing Failure Exit Codes 372

Batch Files 372

Capturing Console Output 372

Keywords 372

Alerts 373

Notification for Alerts 373

Alerts 373

Notification for Alerts 374

Alert Notification Configuration Window 374

Alert Notification Rule 375

Property Pages 375

Identification Page 375

Selection Rules 375

Include messages with these severities 375

Include/exclude message categories 376

Include/exclude specific messages 376

Notification 376

Recipients 376

Minimum interval 377

Subject and Message 377

Notification Severity 377

Jobs Folder 377

Job List 377

Job Details 378

History 378

Job Log 378

Agents 378

Statistics 379

Change Log 379

How To... 379

Create a new Job 379

Move a Job or Group to a different Group 379

Create a new Group 379

View or modify the properties of a Group 380

Delete a Group 380

Queues Folder 380

Notification Recipients Folder 380

Shared Schedules 381

Holiday Definitions 381

Scripts 381

Shared Scripts View 381

Script Libraries View 382

Remote Agents 382

Execution History Query 382

Message Log Query 383

Reports 383

Reports 383

Included Reports 383

Additional Reports 384

Create or Customize Reports 384

Additional Reporting Resources 384

Window Reference 384
Change Log Window 384
Audit Records 385
Snapshots 385
View 385
Compare 385
Restore 385
Clear Job History 386
Clear Job History Window 386
Console Options Window 386
General Options 386
Console Auto-Refresh Interval 386
Startup View 386
Timestamp Display 387
Reset all hidden messages 387
Feature Usage Reporting 387
Credential Profiles 387
Credential Profiles Window 387
Find/Replace 388
Additional Options 388
Security 388
Credential Profiles Security 388
Automatic Permission Granting 388
Available Permissions 389
Credential Profile Properties 389
Property Pages 389
Credentials 389
Applies To 389

Domain 389

User ID 389

Password 390

Comments 390

Use system context for certain operations 390

Agent Profiles 390

Security 391

Automatic Permission Granting 391

Available Permissions 391

Entering User Credentials 392

Deleted Objects Window 392

Duplicate Job Group Window 393

Name for New Group 393

Hold all jobs in new group 393

Prefix new job names with 393

Copy 394

Execute Job Window 394

General Options 394

Ignore conditions for the job 394

Ignore conditions for individual steps 394

Force a new instance of the job if necessary 394

Override queue limits and force the job to run immediately 394

Run only on Master 394

Force job to run on Master 395

Save the selected options as the default options 395

Make the job visible on my desktop 395

Responses 395

Checkpoint 395

Steps 395

Run the job from the beginning 395

Run the job from the specified step 396

Run only the selected steps 396

Variables 396

Notes 396

File Servers 396

File Service Provider 396

File Servers Window 397

Security 397

FTP Service Provider 397

Property Pages 397

FTP Settings 397

Server Type 397

Server Address 398

Port 398

Authentication 398

Use unencrypted control channel 398

Validate server certificate 398

Use Mode Z to compress data 398

Use Passive mode 398

Preserve timestamp on upload 398

Preserve timestamp on download 398

Server time zone 398

Proxy 398

Proxy Type 399

Server Address 399

Port 399

Authentication Type 399

Security 399

Test 400

SFTP Service Provider 400

Property Pages 400

SFTP Settings 400

Server Address 400

Port 400

Authentication 400

Preserve timestamp on upload 401

Preserve timestamp on download 401

Proxy 401

Proxy Type 401

Server Address 401

Port 401

Authentication Type 401

Security 401

Test 402

Holiday Set 402

Holiday Set 403

Holiday Set Properties 404

Property Pages 404

Schedule 404

Name 404

Enable this schedule 404

Rules for all years 404

Rules for specific years 404

Security 404

Job Filter Window 405

Select Jobs and Groups 405

Tags 405

Save settings as 405

Job Flow Diagram 405

Navigating the Diagram 406

Link Types 407

Editing Objects 407

Printing the Diagram 407

Saving the Diagram 407

Job Instance 408

Instance Acknowledgment 408

Job Instance 408

Instance Acknowledgment 408

Instance Details Window 409

Detail Pages 409

Job 409

Instance 409

Status 409

Last Step 409

Last Checkpoint 409

Execution Start 409

Execution Finish 409

Elapsed Time 409

Processor Time 409

Execution Reason 409

Cycle ID 409

Chain ID 410

Log 410

Conditions 410

Steps 410

Captured Files 410

Comments 411

Job Detail Log 411

Step Details 411

Step 411

Description 412

Status 412

Result 412

Execution Start 412

Execution Finish 412

Elapsed Time 412

Processor Time 412

Process ID 412

Last Checkpoint 412

Job Variable Properties 412

Properties 412

Variable Name 412

Hide Value 413

Variable Type 413

Value 413

Description 413

Linked Servers 414

Linked Server 414

Linked Servers Window 414

Upgrading from Previous Versions 414

Linked Server Properties 415

Server to Add 415

The current server can send commands... 415

The target server can send commands... 415

The target computer is not in the same domain as the current computer 416

Override the server address/connection information 416

Network Resource 416

Network Resource 416

Network Resource Properties 416

Network path to connect 417

Assign drive letter 417

User Account 417

Messaging Setup 417

Messaging Setup Window 417

SMTP Servers 417

Incoming E-Mail Servers 418

SMS Gateways 418

Jabber (XMPP) Servers 418

Additional Options 418

Security 418

SMTP Server Properties 419

Description 419

Enabled 419

Server Name or Address 419

Port 419

Return Address for Messages 419

Sender Display Name 419

Maximum Size for Messages 420

Authentication 420

Test... 420

Incoming E-Mail Provider Properties 420

Property Pages 421

Provider 421

Provider Name 421

Server Name or Address 421

Server Port 421

Server Type 421

User ID and Password 421

Test 421

Security 421

SMS Service Provider 421

Finding the Proper Settings 422

SMS Service Provider Properties 422

Properties 422

Provider Page 422

Provider Name 423

Enabled 423

Server 423

User ID 423

Password 423

Maximum Message Length 423

Automatically split messages that exceed the maximum 423

Import Settings 423

Test 423

Security Page 423

Jabber Service Provider Properties 424

Property Pages ... 424

Provider ... 424

Provider Name ... 424

Port ... 424

User ID and Password ... 424

Test ... 424

Security ... 424

Messaging Provider Import ... 425

Notification Recipient Security ... 425

Object Comparison Report ... 426

Object Configuration Report ... 426

Queue Security ... 426

Remote Agent Security ... 427

Resolve Object References Window ... 427

Script Security ... 428

Send Error Logs Window ... 428

Sequence Jobs Tool ... 428

Add sequence number to job names ... 429

Link type ... 429

Server Connection ... 429

Target ... 430

Instance ... 430

Port ... 430

Connection Name ... 430

Authentication ... 430

Windows Authentication ... 430

adTempus Authentication ... 430

Prompt when connecting ... 430

Automatically reconnect whenever the Console is opened 430

Automatically expand the server's node in the Console Tree 431

Make this my default server 431

Server Options 431

Server Options Window 431

Settings Pages 431

Additional Options 431

View server settings change log 431

Advanced Options 431

General Page 432

Data Retention 432

Default History Retention 432

Retain server messages (such as unacknowledged alerts) for __ days 432

Do not delete failed instances that have not been acknowledged 432

Database Operation Task 433

Timeout for Database Operation Task Commands 433

Data Backup Page 433

Create a daily backup of the adTempus database 433

Back up and snapshot each day at 433

Auditing and Snapshot Settings 434

Configuration 434

Auditing 434

Snapshots 434

Retention Settings 435

Variables Page 435

Data Collection and Error Reporting Page 436

Anonymous usage information 436

Diagnostic logs 436

Error Notifications 437

Advanced Server Options 437

Server Security Settings 437

Security Settings 437

Apply Permissions To 437

Logins 437

Include dynamic logins 438

Groups 438

Shared Schedule 438

Shared Schedule 438

Shared Schedule Properties 439

Property Pages 439

Schedule 439

Name 439

Enable this schedule 439

Rules for all years 439

Trigger every ___ days 440

Trigger on specific days 440

Rules for specific years 440

Notes 440

Security 440

Shared Schedule and Holiday Security 441

Text Edit Window 441

Object References Window 443

Tools Reference 444

Data Import and Export 444

Data Format 444

Data Export 445

Object Selection 445

Options 445

Include history when exporting jobs 445

Include passwords for credential profiles 445

Specify a password to protect sensitive data 446

Comments 446

Export Target 446

Export to File 446

File Format 446

Export to Server 446

Data Import 446

Source Page 447

Import data from file 447

Next Step 447

Select Objects Page 448

Special Considerations for Server Settings and Security Containers 449

Next Step 449

More About Duplicate Objects 449

Import Options Page 450

Basic Options 450

Import job history if present in file 450

Import security settings for objects 451

Save all imported objects if no problems are found 451

Generate comparison report for existing objects 451

Create snapshots of existing objects before updating them 451

Show advanced options 451

Server Settings Import 451

Advanced Options 452

Create new copies of all jobs 452

Import group hierarchy under the following group 452

Place imported jobs on hold 452

Next Step 452

Import Results Page 453

Job Templates 454

Job Templates 454

Creating a Template 454

Using a Template 454

Preparing a Job to Be Templatized 455

Use Template Parameters 455

Use Job Variables 455

Remove Notification Recipients 456

Document the Job 456

Create Job Template Wizard 456

Select Job 456

Template Parameters 456

Template Properties 457

Template Instructions 457

Save Template 457

Arcana Scheduler Import 458

Arcana Scheduler Client Components 458

Additional Considerations 458

Select Jobs 458

Import Options 458

Use the following set of holidays 459

Disable imported jobs in the Arcana Scheduler 459

Hold imported jobs in adTempus 459

Successful Imports 459

Failed Imports 459

Create Snapshot Window 459

Find and Replace 460

Search Options 461

Search Jobs 461

Search Jobs 461

Search job history 461

Log messages 461

Captured files 461

Contents of captured files 461

Search Elsewhere 461

Search For 462

Search and Replace 462

Search 462

Replace 462

Go to Job 462

Report Designer 463

Report Designer 463

Open Report Window 463

Create Report Window 464

Report Properties Window 464

Report Title 464

Description 464

Creator 464

Report ID 464

Job Status Descriptions 464

The adTempus Service 466

The Service 466

Managing the adTempus Service 466

Troubleshooting Service Problems 466

Service Account Requirements 466

Limitations When Running the Service Under a User Account 467

Machine ID and Database Ownership 467

Database Ownership 467

Resetting the Machine ID 468

Service Startup Options 468

Available Options 469

Utility Programs 471

Server Tools Launcher 471

adtChkpt Utility 471

Command-Line Job Execution Utility (adtExec) 471

Command-Line Syntax 472

Examples 473

adTempus Database Utility 474

Backing Up the Database 474

Restoring the Database 474

Querying and Updating the Database 475

Engine Mode Utility 475

Resetting an Agent's Master 475

Endpoint Configuration Tool 476

Create Certificate 476

Technical Information 477

Data Security 477

Server Security Certificates 477

Obtaining a Server Certificate 477

Commercially-Issued SSL Certificate 477

Internally-Issued Certificate 477

Self-Signed Certificate 478

Configuring adTempus to Use the Certificate 478

Backing Up and Restoring Data 478

Backing Up Data 478

Restoring Data 479

Port Information 479

Changing the Port 479

Notes 480

Exit Codes 480

How to Set the Exit Code 480

C/C++ Programs 480

.NET 480

Visual Basic 480

Batch Files 481

Scripts 481

Everything Else 481

How Terminates a Process 482

Network Access for Jobs 482

Mapped versus Unmapped Network Drives 482

UNC Paths 482

Drive Letters 482

Using UNC Paths 482

Using Drive Letters 483

Regular Expression Syntax 483

General Rules 483

Single-Character Pattern Matching 484

Lists 484

Repetition 484

Grouping 485

Alternation 485

Predefined Variables 485

Miscellaneous 485

Job 486

Job Step 487

Task: E-Mail 487

Task: File Operation 487

Task: Process Termination 487

Task: Program Execution 488

Task: Script Execution 488

Task: Web Request 488

Trigger: Computer Monitor 489

Trigger: File 489

Trigger: Process 489

Trigger: Event Log 490

Trigger: E-Mail 490

Alert Notification 491

How To... 492

How To: Create a Basic Job 492

Next Steps 493

How To: Send Notification Messages for Failed Jobs 493

How To: Link Jobs Together 494

Linking with a Job Control Action 495

Linking with a Job Trigger 496

How To: Perform an action if a job has not run by a certain time 497

How To: Mirror jobs to Remote Agents 498
Disable Execution for All Jobs 499
Hold the Root job group 499
Shut down adTempus 499
Disable job execution using a startup option 499
Disable job execution using a Registry option 499
How To: Set and Retrieve Variables in Scripts 500
How To: Set the Checkpoint Using the adTempus API 500
How To: Specify Notification Subjects, Messages, and Severities at Run-Time 500
Notification Severity 500
How To: Set Environment Variables with a Script 501

Welcome to adTempus

Welcome to adTempus. Here's some information to get you going.

About adTempus Help

This user manual contains the same content as the online file that is installed with adTempus and is also available at www.arcanadev.com/documentation.

Please visit the online version for the latest content updates and to read or make comments about the documentation.

You are viewing the local help file installed with adTempus.

For the most recent help updates and contributions from other users, try the online (browser-based) of the help system. If you prefer to always use the online help instead of this local help file, you can change the option under **Help> About > Set help source.**

This online help system reflects the latest updates to the adTempus documentation, as well as contributions from users (see the Community Discussion section at the bottom of each page).

If you would prefer to use the help file installed on your computer instead, you can change the option under **Help> About > Set help source.**

We welcome your comments on the adTempus documentation. To send us feedback you can use the link found at the bottom of each page, or post public comments using the Community Discussion feature.

You can download a printable (PDF) version of this user guide at www.arcanadev.com/adTempus/documentation.

Getting Started

If you are new to adTempus, these topics are a good place to start:

Overview of adTempus

Getting Started with adTempus

Additional Help Resources

The following additional resources are available for getting help with adTempus:

- The online (browser-based) of the help system reflects the latest documentation updates, and allows users to contribute comments and notes to each help topic.
- The discussion forums let you ask questions and exchange information with other users.
- The Knowledge Base has information on known software issues, usage tips, and more.

If all else fails, you can contact us for support.

If you prefer to always use the online help instead of this local help file, you can change the option under **Help> About > Set help source.**

What's New in adTempus 4

The following sections describe new and changed features for adTempus version 4. If you are upgrading from adTempus 2, you can find information about other changes since version 2 in the version 3 documentation.

Throughout this documentation, new and updated features are marked with a ★ icon in the left margin.

Be sure to read the information on changes to the security model if you are upgrading, as you will need to review and adjust your security configuration, and possibly create new security logins for users.

Installation and Configuration

adTempus now supports side-by-side installation of multiple versions of the adTempus server software (e.g., version 4 can be installed without uninstalling or upgrading version 3).

adTempus now supports multiple instances of the adTempus server on a single computer, allowing you to run two or more completely isolated adTempus servers on a computer.

Security

The adTempus security model has changed:

- Members of the computer's Administrators group no longer automatically have permission to connect to adTempus. All users must be explicitly defined in adTempus, and adTempus Administrators must be explicitly designated.
- Users are now represented within adTempus by Security Logins, which can be linked to Windows accounts or defined in adTempus with a user ID and password.
- Security Groups (roles) are now created and managed within adTempus, rather than relying on Windows security groups.

See the Security Changes topic for more information.

Job Variables

Job Variables can now be assigned a data type (such as date/time). You can use formatting strings (such as "yyyy-MM-dd" for a date value) when inserting Job Variables using variable tokens.

A new option in the Job Variable definition lets you hide the value of the variable, so it is not visible within the adTempus Console.

A new Variable Selector button () now appears next to all text fields that support Job Variables. This button opens a text edit window that allows you to edit text and insert variables from a list.

The new Job Variable Update Task and Job Variable Update Action let a job easily set update a Job Variable. The update can be applied to the executing instance of the job, or it can update the variable definition at the Job, Group, or Server level. The related new Job Variable Condition lets you set a condition on the value of a variable.

The Job Condition now has an option to a Job Variable when selecting a target job instance (see here).

Inline function scripts allow you to call a function in a script and insert the result anywhere you can use a Job Variable. For example, you can use the syntax `%=GetDate("yyyy-MM-dd")%` to insert the current date at runtime.

Jobs

The **Enabled** option and related Hold/Release commands, and the **Allow job to be triggered by other jobs** option, have been replaced with a new Hold Status that allows you to allow or block three kinds of execution separately: scheduled (triggers), on demand (manual), and execution by Job Control Actions. The option to run a job even if it is held (previously available in the execution options for a Job Control Action and when executing a job on demand) has been removed: it is not possible to override the job's Held status.

The User Interaction settings have been simplified. The option to run a job in the console session has been eliminated.

Jobs can be assigned user-defined tags, which can be used to filter jobs in the new Job Monitor view.

New Response Events allow you to execute Responses when:

- A trigger for the job failed
- Execution time reached a threshold
- Scheduled execution missed

The Sequence Jobs Tool simplifies linking and reordering jobs that are connected using Responses.

Job Groups

You can now configure Responses at the group level. Responses defined for a group apply to all jobs within the group and its sub-groups. For example, you can configure the same failure notification rules to be used by all jobs within the group.

Job Groups can now be configured to use a Cycle ID, which gets attached to all jobs that run during a single processing cycle. This makes it easier to configure Job Conditions to target only jobs that ran in the same cycle.

Job Groups now have the same Hold Status option as Jobs. Use this option to temporarily block execution of all jobs in the group. Unlike the previous option to hold or release all jobs in the group, this option does not permanently change the settings for the individual jobs. That is, when you remove the Hold status for the Group, any jobs that were on hold before you held the group remain on hold.

Job Queues

You can now configure Responses at the queue level. Responses defined for a queue apply to all jobs within the queue.

Job Queues now have the same Hold Status option as Jobs. Use this option to temporarily block execution of all jobs in the queue.

Tasks

Program Execution Task

The Program Execution Task now allows you to execute a batch file that is stored in adTempus. The adTempus script editor is used to edit the batch file from with adTempus.

Users who are currently using the Program Execution Task to call an external batch file should consider importing their batch files to take advantage of this new feature. Since the batch file is stored within adTempus, you do not need to worry about deploying the batch file to agents, and changes to the batch file can be tracked using the adTempus auditing and snapshot features.

The batch file can also call shared batch files, which are managed from the Shared Scripts view. This allows for reuse of batch file code.

Users who in the past have created jobs with a large number of steps that each execute a Program Execution Tasks may want to use internal batch files instead, if you are not using conditions or responses on individual steps. This makes it easier to rearrange the order of programs, or to copy and paste command lines between programs.

Using checkpoints you can still get feedback about the progress of the batch file and start or restart a batch file from a particular location.

Database Operation Task

The SQL Server Job Task available in version 3 has been replaced with the Database Operation Task, which supports running jobs, querying, and executing SQL against SQL Server, Oracle, and MySQL databases.

E-Mail Processing Task

The new E-Mail Processing Task can be used to retrieve e-mail messages from a POP3 or IMAP server and save the message body or attachments for further processing.

File Transfer Task

The new File Transfer Task allows for copying, moving, and deleting files. It supports local files and FTP and SFTP servers.

File Compression/Uncompression

The new File Compression and File Uncompression tasks allow for creating and uncompressing ZIP archives.

Job Variable Update Task

The new Job Variable Update Task allows jobs to easily update Job Variable definitions.

Computer Restart Task

The Computer Restart Task now support shutdown/restart of remote computers and can be configured to work even if the user account used for the job is not an Administrator account.

Service Control Task

The Service Control Task now allows you to start and stop services on remote computers and can be configured to work even if the user account used for the job is not an Administrator account.

Notification Task

The Notification Task can now be used to send messages to ad-hoc e-mail addresses without configuring them as Notification Recipients in adTempus.

Conditions

The Job Instance Properties window now shows the status of all conditions for an active or completed instance. While a job is waiting for conditions to be met, this allows you to check to see what conditions have not been met. After a job has run, you can use this information to see, for example, which file caused a File Condition to be met.

Job Condition

The Job Condition has several new rules for selecting a target job instance. Details

The following new instance rules have been added:

- Most recent since previous successful execution of current job. adTempus will look at the most recent instance of the target job that executed since the last *successful* execution of the dependent job.
- Most recent in cycle. adTempus will look at the most recent instance of the target job that has the same Cycle ID as the dependent job.
- Most recent instance since date/time specified in a Job Variable. adTempus will retrieve the current value of the specified variable (which must be defined as a Date/Time variable) and then look at the most recent instance of the target job that executed since that date/time.
- Any instance since previous successful execution of current job. adTempus will look at all instances that have completed since the last *successful* execution of the dependent job. If any of these instances matches the Rule, the condition will be satisfied.
- Any instance in cycle. adTempus will look at all instances of the target job that have the same Cycle ID as the dependent job.

- Any instance since date/time specified in a Job Variable. adTempus will retrieve the current value of the specified variable (which must be defined as a Date/Time variable) and then look at all instances of the target job that have executed since that date/time.

In addition, the Cycle ID concept has been introduced as a way to link jobs that are running as part of the same processing cycle, making it easier for a condition to target an instance from the same day's cycle as the dependent job.

Two new target condition rules have been added: "Job has not run" and "Job has not run successfully." These rules can be used to have one job take action (such as send a notification message) if another job has not run by the expected time. Example.

Job Variable Condition

The new Job Variable Condition lets you set a condition on the value of a variable.

Triggers

You can now define Job Variables at the Job Trigger level, allowing you to set different variables for a job based on how it is triggered.

E-Mail Trigger

The new E-Mail Trigger can be used to retrieve e-mail messages from a POP3 or IMAP server and trigger jobs based on properties of the message (such as keywords in the message, or specific kinds of attachments).

File Trigger

The File Trigger can now watch for files on FTP and SFTP servers.

A new option allows adTempus to use the file's last modification to determine whether the file is still being updated by another application.

The interval at which adTempus checks for file creations/deletions/modifications is now configurable.

A new option allows you to configure adTempus to wait on more than one file using "and" logic (e.g., trigger when file A and file B have both been created).

If target files are on a removable or network drive, adTempus now has better handling for scenarios where the drive is removed or the network connection is lost. Previous versions treated this scenario as a file deletion (because the target files were not found). The trigger now waits until the drive is present again and compares the new state of the file system to the state before the drive was disconnected.

Job Trigger

The new Job Trigger simplifies the linking together of dependent jobs. It functions similarly to the Job Control Action, except that it is configured on the job to be run, not on the dependency job. It also allows the job to be configured to wait for more than one to complete, providing some of the same capability of the Job Condition.

Schedule Trigger

Shared Schedules and Holiday Sets can now have rules that apply only to specific years.

The start time for scheduled jobs can now be randomized within a set limit to ease the load on adTempus or the server. For example, if you have a large number of jobs that run every hour, you can configure them to start randomly within 5 minutes of the beginning of the hour.

The Day Selection window has been reorganized to be easier to use.

The floating day rule has been enhanced to include "on or before" and "on or after" options in addition to the existing "before" and "after" options.

The weekday rule now allows you to skip weeks (e.g., every third Monday).

The predefined "weekday" (Monday through Friday) and "weekend" (Saturday, Sunday) options have been removed from the weekday and floating rules; the floating day rule has been enhanced to support alternative definitions for the work week. For example, you can now schedule to run a job on the 5th business day of the month where business days are defined as Sunday through Thursday rather than Monday through Friday.

Scripting

The script hosting process has been improved to provide better performance, especially when many scripts are being executed at the same time.

adTempus natively supports **Windows PowerShell** scripts, which are stored and edited within adTempus just like other scripts.

New inline function scripts allow you to call a function in a script and insert the result anywhere you can use a Job Variable. For example, you can use the syntax `%=GetDate("yyyy-MM-dd")%` to insert the current date at runtime.

Script Libraries written using .NET languages (VB.NET and C#) can now be shared across languages (e.g., a C# script can reference a VB.NET library).

The base class for scripts written using .NET languages now exposes a Parameters object, which provides simplified access to information previously passed to scripts through Job Variables. For example, a script run to select files for a File Trigger can get a list of the files found by the trigger.

Scripts can now send e-mail messages and capture files to the file history.

When logging messages using the adTempus.LogMessage method, scripts can use the MessageTypeEnum.Debug logging level to send messages to a captured log file instead of writing to the Job Log. This option should be used when generating debug/detail messages or in other scenarios where a large number of messages is being logged for a script. This avoids filling up the job log database table with messages, which can slow console performance.

Auditing, Change Tracking, and Snapshots

Auditing (logging of changes made to objects) was previously only available for jobs but has now been extended to cover all adTempus objects (groups, queues, shared scripts, etc.).

adTempus can now be configured to take a snapshot of an object before it is updated. This allows you to later compare versions of the object, view the settings from an old version, or restore the object to its earlier state. This also allows for recovery of deleted objects.

adTempus can be configured to make a daily system-wide snapshot of all objects to augment the database backup. The snapshot allows for granular restoration of one or a few objects, whereas the database backup can only be used to restore the entire database.

See the Auditing and Snapshots Overview and the Auditing and Snapshots Settings for more information

E-Mail Notification for Alerts

The Alerts view now allows you to configure notification rules for alerts. For example, you can configure a rule to notify an administrator when adTempus reports an operational problem such as a database connection problem, or when a Remote Agent connection is lost.

Job Monitor View

The new Job Monitor view in the Console provides a view of recent, active, and upcoming jobs.

Distributed Scheduling Changes

The performance and stability of the Distributed Scheduling subsystem (Master/Agent communications) has been greatly improved, and some new features in the Remote Agents view make administration of Agents easier:

- The list of agents now shows the version of adTempus installed on each Agent, and indicates any agents that are not running the same version as the Master.
- The Configuration Report command produces a configuration report for the selected agents, showing the same server configuration information that appears in the new About box.
- The Master can now push software updates to agents.
- The Resynchronize command removes all replicated configuration data from an Agent and re-sends all required data from the Master.

Credential Profiles can now be configured to use different credentials for one or more Agents. For example, if you are running jobs under a local user account instead of a domain account, or if an Agent is in a different domain, you can specify a different set of credentials to use when the job runs on that agent. See the Agent Profiles Page in the Credential Profile window for more information.

Other Console Improvements

Console performance/responsiveness have been greatly improved.

Many property windows (such as editing jobs) are now "modeless," meaning that you can have more than one open at a time.

Many grids (such as job and instance lists) now support column reordering and also allow you to choose which columns are displayed. Many new optional columns have been added. Right-click a column header to display a list of the available columns.

Search and Replace have been enhanced to include most text fields throughout adTempus. It is now possible to search the contents of captured files for jobs.

It is now possible to find/replace object references. For example, you can find all locations where a Credential profile is in use and replace the references with a different Credential Profile.

When a software update is installed on the adTempus server, Consoles installed on remote computers can download updated software from the server.

Export/Import

Export/Import now supports exporting/importing any top-level object (Notification Recipient, Script, etc.) without exporting a job that contains it. It is now possible to export and import server settings as well.

You can now specify a custom password to use when encrypting sensitive data (like passwords) in an export file, providing for greater security.

When importing jobs, you can choose to import all jobs as a copy. When you do this, adTempus generates a new identity value for each job so that they do not replace existing jobs. More information.

You can also choose to recreate the group structure of the import source under a different target group. For example, if you are importing jobs created on another server and want them all placed under a Job Group named "Imported Jobs," adTempus will recreate the group hierarchy from the source underneath this target group. More information.

The export file format has changed from the version 3 format, but adTempus 4 is able to read and write the version 3 format.

Job History

A new Job Detail Log created for each job instance contains various details from the job execution. The log shows the values of all Job Variables at the time of execution, and includes detailed informational messages about some operations. For example, the File Transfer Task reports here information about all the files that it transfers. Some informational messages that were previously reported to the Jog Log have been moved to the Detail Log.

Some informational messages have been removed or moved from the Job Log to the Detail Log to reduce the number of messages in the Job Log.

The informational message that gave the execution reason for each instance has been eliminated; the execution reason is now displayed in the Instance Details window.

Server Trust Relationships

In order for a Job Condition, Job Trigger, or Job Control Action to target another adTempus server, you must first establish a trust relationship between the servers using the Linked Servers window. If you are upgrading from a previous version, adTempus will log an Alert with ID 5270 at service startup if it finds remote targets for which no Linked Server definition has been created. You will need to configure the trust relationship before the condition or response can work.

New Reporting Platform

The reports included in the adTempus Console have been converted to a new reporting technology, and a new report designer tool makes it easier to customize reports or create new ones.

Miscellaneous Changes

The Process Termination Task and Program Execution Task now terminate the entire process tree when terminating a process. In previous versions of adTempus, if you terminated a batch file, for example, the batch file was terminated but any programs started by the batch file continued running. adTempus will now terminates processes started by the batch file.

Credential Profiles can now be configured to issue a reminder alert when the password needs to be changed.

Credential Profiles can be configured to use the adTempus system security context when executing Service Control Tasks and Computer Shutdown Tasks, allowing you to execute these operations without using an Administrator account for jobs.

A temporary directory is now created for each executing job, with the path available in the "ADTJobTemp" Job Variable. Tasks and scripts can use this directory to store temporary files, and adTempus will delete them when the job completes.

A quick Configuration Report can now be generated for top-level objects (Jobs, Groups, Notification Recipients, etc.).

New Response Events for jobs:

- Execution time exceeded a threshold value
- Scheduled execution missed because adTempus was not running
- A Trigger for the job failed
- Job queued for execution longer than a specified time
- Job or step waiting for conditions longer than a specified time

An improved About box (**Help > About**) includes various bits of technical information about the adTempus server.

The network broadcast notification address type has been removed from Notification Recipients, as the Messenger service is no longer supported as of Windows Vista/Windows Server 2008.

Contacting Us

Sales

For information on adTempus pricing, to purchase a license, or for sales-related questions, please visit www.arcanadev.com/shop or contact us by e-mail at sales@arcanadev.com.

Technical Support

For technical support information please visit www.arcanadev.com/adtempus/resources.aspx.

Product Feedback

If you have comments about a specific topic in this help document, please use the feedback link found at the bottom of the topic page. This ensures that we receive the information we need to identify the topic.

For general comments about the help document, you can use the form here.

For other comments, including feature requests, please use the form here.

Introduction

Overview

adTempus is a batch process scheduling and automation tool for Windows platforms.

By "batch process" we mean a program (or collection of programs, scripts, etc.) that are meant, for the most part, to be run in an "offline" or unattended fashion, such as a process that reads data feeds and updates your data warehouse each day. In adTempus, such a process is represented by an object called a **job**, which stores all the information adTempus needs to run the process.

By "scheduling and automation" we mean that adTempus executes these jobs at specific times or in response to other external events, and manages dependencies within and between jobs.

The **adTempus Server** (or "engine") is the part of adTempus that is responsible for executing jobs, by running programs and scripts, downloading files, sending e-mail notification messages, etc.

The **adTempus Console** is the primary user interface and management tool for adTempus. Using the Console you manage adTempus servers and the jobs defined on those servers. The Console can run on a different computer than the Server, allowing remote administration of adTempus Servers.

Related Topics

Getting Started 129
Key Concepts 81

Key Concepts

Introduction to Key Concepts

This topic provides an overview of the architecture of adTempus, introducing key concepts and components of the software.

adTempus is highly modular in design. This means that there are many pieces (we will call them "objects"), most of which are fairly simple, that can be combined in many ways to produce complex configurations.

Before we proceed, we will outline an example that will be used to illustrate concepts as we go.

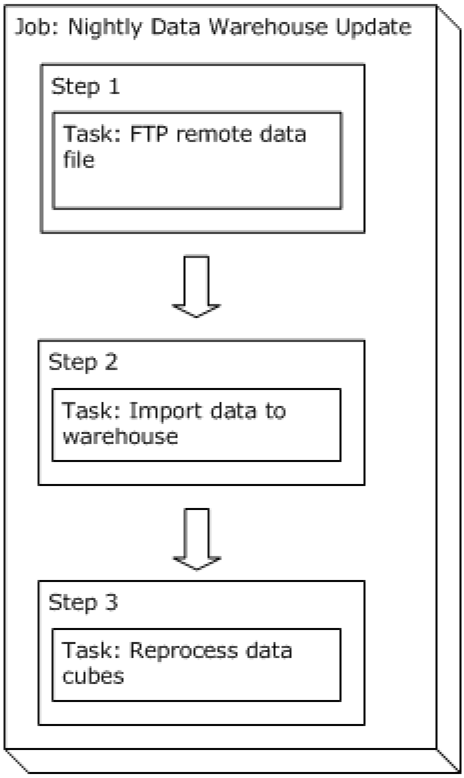
Example

You need to automate the nightly update of your organization's data warehouse. This involves several different steps:

1. Download a data file using FTP.
2. Load the downloaded data into the data warehouse.
3. Run the process that reprocesses the data warehouse and rebuilds the data cubes.

Note that each of these three steps involves a single discrete task.

Jobs, Steps, and Tasks


Job: Nightly Data Warehouse Update
Step 1
Task: FTP remote data file
Step 2
Task: Import data to warehouse
Step 3
Task: Reprocess data cubes

Because adTempus is a job scheduler, the central object is the **Job**. In adTempus the job contains all of the information needed for adTempus to accomplish a specific goal for you. Each job contains one or more **Step**; each step executes a single **Task**.

In our example above, our goal is to complete the nightly data warehouse update; this becomes our job. Each of the three steps we listed above becomes a step within the job.

Note that in some other automation tools, "job" and "task" are synonymous: each of the steps above would be a separate job and the jobs would be linked together to complete the entire process. While you could set up your processing flow this way in adTempus, it is often easier to manage fewer jobs with more steps than more jobs with fewer steps. For more information, see the Multiple Steps vs Separate Jobs topic.

The task represents a single operation that adTempus is to perform (typically, a single program, script, batch file, or other operation that it is to execute). adTempus supports several different kinds of tasks that allow you to run programs, scripts, and batch files; launch documents; restart the computer; control services; and more. For more information, see the Tasks topic.

For more information, see the Jobs and Tasks topics.

Triggers

Jobs can be started manually (a user tells adTempus to run a particular job) but are generally started automatically by adTempus at specified times, or in response to other events on the computer.

A **Trigger** is what tell adTempus when to run a job. For example, if you want your data warehouse update to run at 11:30 each night, you would add a schedule trigger to the job to trigger it at 11:30. Other kinds of triggers allow you to create jobs that will run in response to other events, such as when a file is uploaded to a particular directory on your server.

For more information, see the Triggers topic.

Conditions

Conditions place restrictions on whether a job or step should run. For example, you can specify that a job should run only if a particular file exists.

Conditions do not cause a job to start—only triggers do this. Conditions are evaluated *after*a trigger starts a job.

For more information, see the Conditions topic.

Responses, Events, and Actions

Responses allow you to respond to state changes during the execution of a job, and to customize the flow of execution within and between jobs.

At various points during the execution of jobs and steps, adTempus fires Events. For example, an event is fired when a job starts, when a step starts, when a step finishes successfully, if a step fails, etc. To each of these events you can attach Responses, which specify Actions adTempus should execute when the event occurs. Actions allow you to execute other jobs or steps, send notification messages, capture files, and run scripts.

Using a Response, for example, you can send an e-mail notification message when a job fails or start the next job in a sequence when the job succeeds.

For more information, see the Responses topic.

Job Groups

Job Groups allow you to organize your jobs into "folders" for easier management in the Console. Job Groups are simply a visual organizational tool, and do not create any sort of relationship between the jobs in a Group.

For more information, see the Job Group topic.

Job Queues

Job Queues allow you to limit the number of jobs that can run at the same time, or prevent certain jobs from running at the same time.

For more information, see the Job Queue topic.

Related Topics

Introduction 81
Getting Started 129
adTempus Objects 133
Conditions 258
Triggers 212
Tasks 147
Steps Overview 145

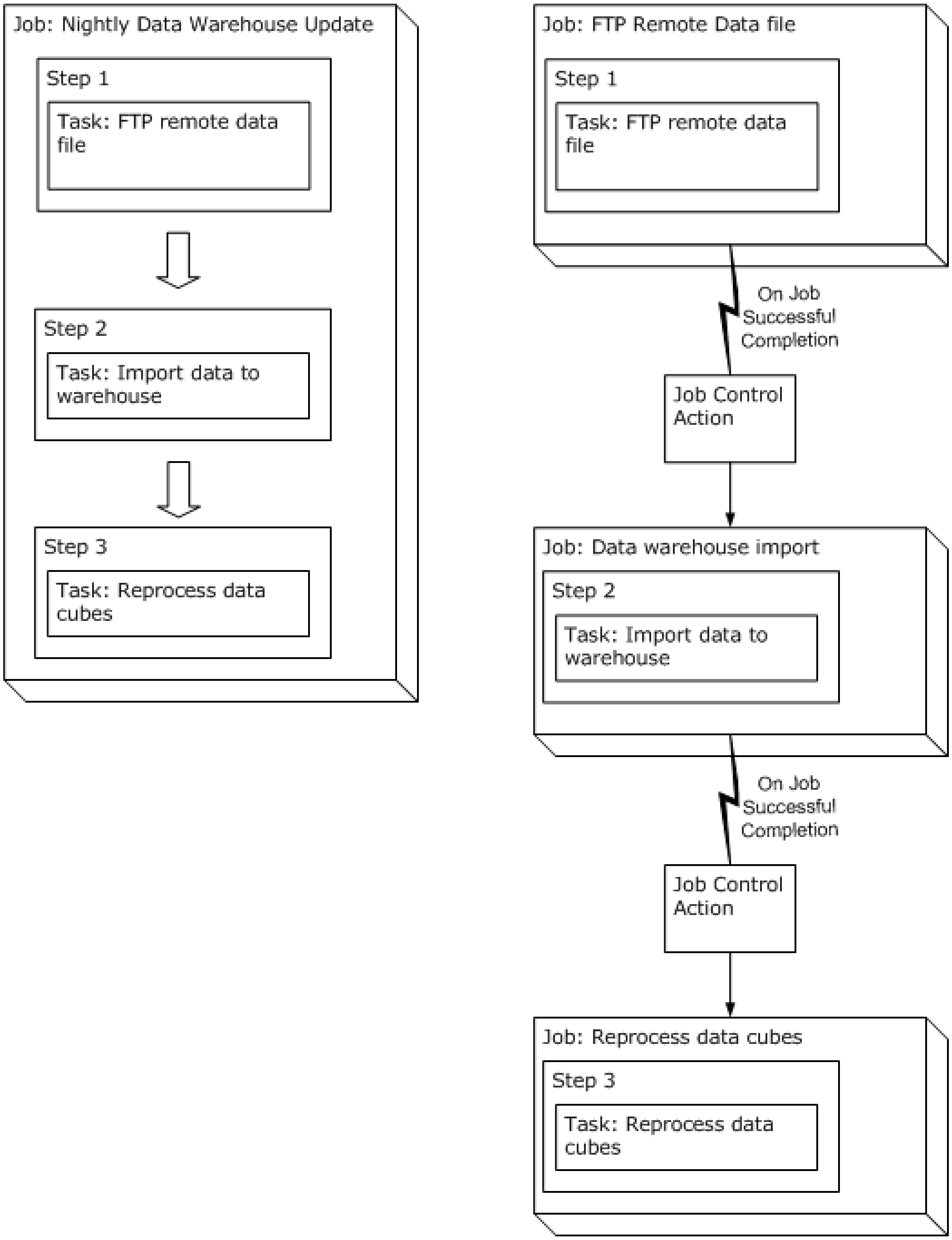
Multiple Steps vs Separate Jobs

Generally, if two tasks will always be executed together in the same sequence, they should be part of the same job. If the tasks sometimes need to run separately, or in a different order, or as part of a different sequence, they should be in separate jobs.

adTempus allows you to group related tasks into a single job, using multiple steps within the job. Grouping tasks this way simplifies administration of adTempus, because you have fewer jobs to administer, and the relationship between the tasks is clear.

In most cases the two approaches are equivalent and interchangeable. For example, the following setups both accomplish the same result:

Multiple steps within the same job	Separate jobs, connected using Job Control actions


Job: Nightly Data Warehouse Update
Step 1
Task: FTP remote data file
Step 2
Task: Import data to warehouse
Step 3
Task: Reprocess data cubes
Job: FTP Remote Data file
Step 1
Task: FTP remote data file
On Job Successful Completion
Job Control Action
Job: Data warehouse import
Step 2
Task: Import data to warehouse
On Job Successful Completion
Job Control Action
Job: Reprocess data cubes
Step 3
Task: Reprocess data cubes

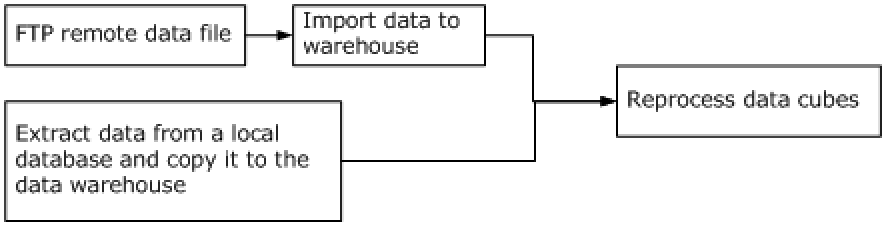
There is, however, one key difference between steps and jobs that may require you to use multiple jobs: steps within a job execute sequentially, one at a time. That is, within a job only one step may be executing at any given time. Jobs, on the other hand, can execute in parallel: any number of jobs can be executing at once.

To illustrate the ramifications we'll add one more task to our data warehouse update example. We currently have the following steps:

1. Download a data file using FTP.
2. Load the downloaded data into the data warehouse.
3. Run the process that reprocesses the data warehouse and rebuilds the data cubes

Now we'll add a new task before task 3:

1. Download a data file using FTP.
2. Load the downloaded data into the data warehouse.
3. Extract data from a local database and copy it to the data warehouse.
4. Run the process that reprocesses the data warehouse and rebuilds the data cubes.


FTP remote data file
Import data to warehouse
Extract data from a local database and copy it to the data warehouse
Reprocess data cubes

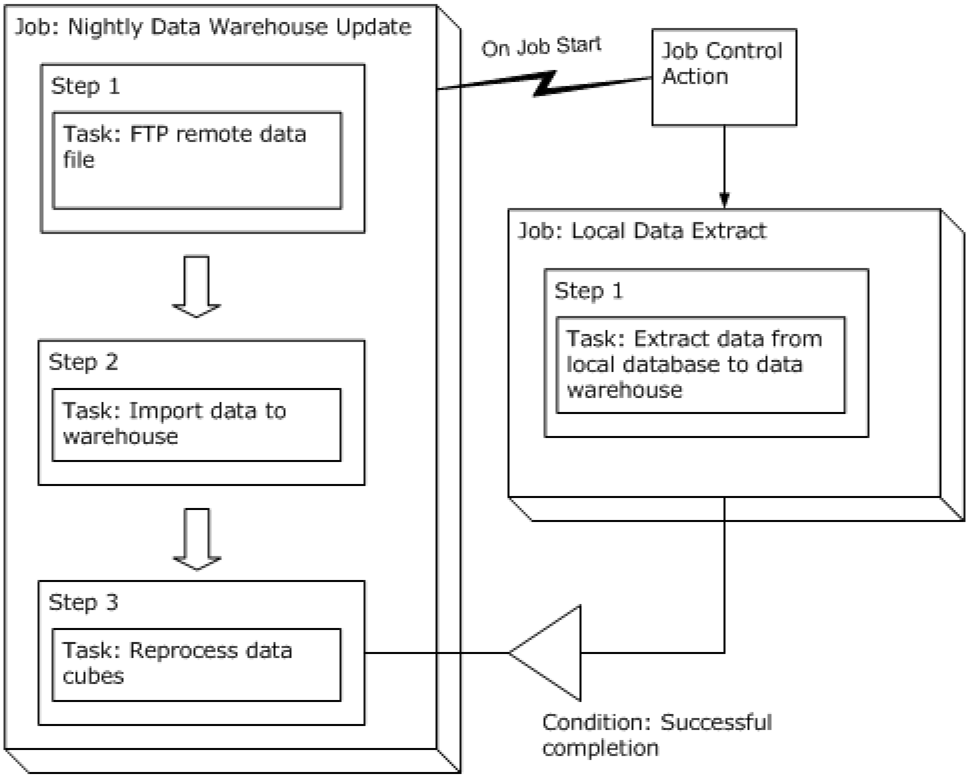
We have the following dependencies among the tasks:

- Task 2 must follow task 1.
- Task 3 can execute independently of tasks 1 and 2.
- Task 4 must follow tasks 2 and 3.

Since task 3 is not dependent on tasks 1 and 2, it makes sense to have it execute concurrently with tasks 1 and 2—there's no need to wait until they're finished.

To do this, we must move this task to a separate job. We then link the two jobs together as follows:

- An action attached to the "Job Start" event on our original job starts the local extraction job. That is, whenever the main job is started, the second job is started as well.
- Step 3 of the main job does the data warehouse reprocessing. Step 3 will be automatically started once Step 2 (which imports the downloaded data) completes. However, we need for Step 3 to wait until the local data extract (executing in the secondary job) completes. So we add a Job Condition to Step 3, causing it to wait until the secondary job completes.


Job: Nightly Data Warehouse Update
Step 1
Task: FTP remote data file
Step 2
Task: Import data to warehouse
Step 3
Task: Reprocess data cubes
On Job Start
Job Control Action
Job: Local Data Extract
Step 1
Task: Extract data from local database to data warehouse
Condition: Successful completion

Comparison of Triggers and Conditions

Users who are new to adTempus sometimes have trouble knowing when to use Triggers and when to use Conditions, and often use Conditions when they should be using Triggers.

The key point to remember is that Triggers are "**active**," while conditions are "**passive**." This means that a Trigger can cause a job to run. A Condition can never "cause" a job to run; it just determines whether a job (or step) is allowed to run when something else triggers the job. Conditions are never evaluated until the job has been triggered (started) by a trigger, manual execution, or a Response on another job.

For example, adTempus includes both a File Trigger and a File Condition, which have similar options and features. When you add a File Trigger to a job, the job is triggered (started) when files are created, modified, or deleted (depending on the File Trigger settings). When you add a File Condition to a job, the job still must be triggered by some other means. Once the job is triggered, the File Condition is used to determine whether the job should continue running.

The confusion between Triggers and Conditions appears most often when a user wants a job ("Job B") to run when another job ("Job A") finishes. Users often try to configure this by giving "Job B" a Job Condition on "Job A." Because the condition is passive, though, it has no effect unless something else triggers "Job A." For this scenario you would use a Job Trigger or a Job Control Action instead.

Related Concepts

Triggers 212
Conditions 258

Job Variables

Job Variables allow you to easily use changeable values in the configuration and execution of your jobs. Job Variables are similar in concept and usage to Windows environment variable: you create a variable, giving it a name and a value. You can then insert the variable (by referencing its name) in various places within adTempus (such as the command line for a program to be executed). When adTempus runs the job, it replaces the reference to the variable with the variable's current value.

For example, supposed you have a job with 8 steps. Each of those steps runs a program that requires as a command-line parameter the name of the folder from which it should read data files. All of the steps use the same folder, which is usually `c:\clientdata\inputfiles`. However, occasionally you need to do a special run of this job, reading data from a different folder. Without Job Variables, such a change would require you to edit all 8 steps to change the command-line parameters to point to the new location.

With Job Variables, you can create a variable for the job (on the Variables page of the job's properties) named "DataPath" and set the variable to `c:\clientdata\inputfiles`. In each Job Step, instead of putting the path `c:\clientdata\inputfiles` in the command-line parameters, you put a reference to the job variable, using this syntax: `%DataPath%`.

When the job runs, adTempus will replace `%DataPath%`" with the current value of the "DataPath" Job Variable.

Now when you need to do a special run using a different path you can simply change the value of the "DataPath" variable in one place (in the job's properties) and the new value will be used throughout the job. And if you run the job manually, you can override the "DataPath" variable at runtime without even modifying the job (using the Variables page of the Execute Job window).

You can also use Inline Functions anywhere you can use a Job Variable. Inline Functions let you execute script code and then insert the result.

Job Variable Inheritance

Job Variables may be defined at many levels within adTempus: Server, Job Group, Job Queue, Job, Step, and Trigger. Variables set at a level are inherited by all objects at a lower level: variables set at the Server level are inherited by all Groups and Queues on that server, variables set for a Group are inherited by all Groups and Jobs within that Group, etc.

Inherited Variables can be overridden to change their values. In the previous example the "DataPath" variable was defined for a job. This variable will now appear on the "Variables" page in the properties of each step in the job. If one step needs to use a different value, you can edit the variable for that step only and give it the necessary value.

Using Job Variables

Job Variables can be used throughout adTempus in most places where you enter text that is used in the execution of a job. For example:

- The execution target and command-line parameters for a task.
- Paths for file triggers and conditions.
- Notification message subjects and bodies.

Text entry boxes that support Job Variables have an Insert Variables button (▤) next to them. This button opens the Text Edit Window, which allows you to select variables from a list and insert them into your text.

You can also insert variables by directly typing the name of the variable enclosed in percent signs, like this: `%Variable Name%`. Variable names are not case sensitive, so you could also enter the variable as: `%variable name%`.

Variables can also be retrieved and set within scripts and set using the Job Variable Update Task or Job Variable Update Action.

Variable Formatting

For date, time, or numeric variables, you can use standard .NET formatting strings to format the value when you insert it somewhere using a token. For example, suppose you have defined a variable named "ProcessingDate" that you set to the current date at the beginning of your processing cycle. You need to include this value in the command line parameters for several programs that you run, but some of the programs need the date to be formatted differently than others. You can insert the date in `yyyy-MM-dd` format using this token: `%ProcessingDate{yyyy-MM-dd}%`. If another program needs the date formatted as `MM/dd/yyyy`, you can insert it using this token: `%ProcessingDate{MM/dd/yyyy}%`.

Formatting like this only works if you set the Variable Type properly when you define the Job Variable.

Related Topics

How To: Set and Retrieve Variables in Scripts 500
Job Variable Condition 269
Job Variable Update Action 289
Job Variable Update Task 178
Predefined Variables 485
Text Edit Window 441

Reference

Job Variable Properties 412

Resources

Resources represent system resources that must be allocated to a job before it can run. adTempus supports the following resource types:

- The Network Resource allows you to map network connections for the job.

Related Topics

Network Resource 416

Reference

Job Resources Page 141

Cycle ID

This feature is new for adTempus version 4.

The **Cycle ID** is used to track the execution cycle that job instances belong to.

For example, suppose that each day you run a set of jobs that are linked together using Job Conditions or Job Triggers. One of these jobs, "Data Copy," uses a Job Condition to ensure that it does not run until job "Data Extract" has run successfully. You want to make sure that when "Data Copy" looks to see if "Data Extract" has succeeded, it is looking at the instance that ran in today's cycle, and not the instance from yesterday's cycle.

In earlier versions of adTempus you would enforce this by using a rule that told adTempus to only look at instances of "Data Extract" that ran since "Data Copy" last ran. This rule works in most cases, but can cause problems if jobs get run out of sequence or out of cycle.

The Cycle ID concept allows you to define a unique identification tag that gets assigned to all job instances run during a particular time period. That way you can easily tell the "Data Copy" job to only look at instances of "Data Extract" from the same cycle, without having to rely on comparing execution times.

How the Cycle ID is Assigned to a Job

When a job is submitted for execution, adTempus sets the Cycle ID for the instance to the current Cycle ID defined for the job's Cycle Scope (see next section).

The Cycle ID is simply a timestamp in the form `yyyy-MM-dd HH:mm:ss`.

Cycle Scopes

When looking for the Cycle ID, adTempus starts with the Job Group to which the job belongs and then works its way up through the group hierarchy, stopping at the first group that has a Cycle ID configured. This means that you can have different Cycle IDs active for different "scopes" as defined by job groups. To define a new scope, simply configure the Cycle ID options in the group that represents the base of the scope.

For example, suppose that under the root job group you have two main Job Groups, "Data Warehouse Updates" and "Customer Data Processing." Each group's jobs are related to a particular business area and run on different processing cycles. Therefore you do not want to use the same Cycle ID for both groups, because resetting the cycle ID at the beginning of one group's cycle would change the Cycle ID in the middle of the other group's cycle.

To create two different Cycle ID scopes, simply enable the Cycle ID options (in the Job Group properties) for each of the groups. All jobs that run in a group (or any of its descendent groups) will get the same Cycle ID.

If all your jobs work as part of the same daily cycle, you can just set the Cycle ID options for the root job group.

How to Update the Cycle ID

For the Cycle ID to be useful, it has to be updated to a new value at the start of each cycle. To do this, identify the job that runs first in your cycle and check the **Update the Cycle ID for this job's scope when the job runs** option on the Trigger page of the job's properties. Each time this job runs, a new Cycle ID will be generated for the scope that contains the job.

Viewing the Cycle ID for an Instance

To view the CycleID for an active or completed job instance, you can display the properties for the instance. The Cycle ID will be listed on the **Job** page of the properties window.

The Cycle ID is also available as a column in the History panel, Job Monitor view, and Failed Jobs view. If it is not already displayed, right-click an existing column heading to bring up a list of the available columns, and check the Cycle ID option.

Credential Profiles

Credential Profiles simplify the use and management of user credentials in adTempus. Each Credential Profile stores the user ID and password for a single user account. In most cases this will be a Windows user account, but you may also need to store other kinds of credentials, such as the credentials for connecting to a protected Web server.

Instead of storing user IDs and passwords with each object that uses them (for example, in each job that runs under a user account), adTempus stores the information in one place (the Credential Profile) and then stores a reference to the Credential Profile in each object that uses it.

Features and Benefits

Credential Profiles offer several benefits.

Simplified Password Management

If the password for an account is changed, the password only has to be changed once in adTempus, in the Credential Profile. The new password will immediately be used by all jobs that use the profile.

Delegated Account Use for Improved Security

Users can be granted permission to use a Credential Profile without knowing the password for the account.

For example, it is often necessary that jobs be run under user accounts with Administrative privileges, and non-administrative users may need to configure and manage these jobs. Instead of telling these users the password for an administrative account to use on jobs, an authorized Administrator (who knows the account's password) can create a Credential Profile, and grant other users permission to use the profile. These users will now be able to schedule jobs using this account without having access to the account's password.

Simplified Data Entry

Once a user has been granted permission to use a Credential Profile, that user does not have to type in (or even know) the password for the profile. When entering credentials for a new job or other object, the user only has to type the user ID.

Creating and Using Credential Profiles

Generally, Credential Profiles will be created automatically, as needed. In places where credentials need to be specified (such as the **User Account** box in a job's properties), the user simply types in the user ID for the account. adTempus will then prompt the user for the password and create a new Credential Profile if necessary, as described in the Entering User Credentials topic.

You can also create profiles through the Credential Profiles window.

Managing User Profiles

Authorized users can view and manage Credential Profiles through the Credential Profiles window.

If the password changes for an account, use the Credential Profiles window to edit the associated Credential Profile and change the password in adTempus.

Changing Credentials

If want to change all jobs that use a particular Credential Profile to use a different user account, you can simply change the user name and password in the profile. For example, suppose you have a Credential Profile for the account "corpnet\automation." You have moved the adTempus server to a new domain, and the jobs should now run under account "newcorpnet\autouser." You can make this change by simply editing the Credential Profile and changing the user ID and password. All jobs that use the profile will now use the new credentials.

If you only want to change *some* jobs that use a Credential Profile to use a different profile (while others will continue to use the old profile), use the **Find/Replace References** command from the Credential Profiles window.

Related Topics

Entering User Credentials 392

Reference

Credential Profile Properties 389
Credential Profiles 387

Notification Recipients

adTempus features two kinds of notification recipients. A Notification Action, File Capture Action, or Notification Task may send notification messages to any number of recipients or either kind.

- A Notification Recipient, or individual, represents a single person or entity who receives notification messages. Each recipient may have any number of addresses, which may be used in different circumstances. For example, an individual may have an e-mail address that receives all notification messages sent to that person, and a smart phone e-mail address that only receives severe notification messages.
- A Notification Group is a collection of Notification Recipients or other groups. Each group can have any number of members, and schedules can be used to determine which members receive notification at certain times. For example, you may have a group that represents your on-call duty schedule, where one person receives notification messages that are sent between 9 AM and 6 PM, and another person receives messages from 6 PM to 9 AM.

Related Topics

Notification Task 181
Notification Action 291
Messaging Setup 417
Notification Recipient 303
Notification Group 309
How To: Send Notification Messages for Failed Jobs 493

Using Scripts in adTempus

Scripts are small "programs" that you write in various programming languages to carry out a task or add functionality to adTempus.

Scripts are represented within adTempus by the Script object, which allows you to share scripts among jobs and users, and to restrict the use of scripts with security settings.

Scripts can interact with adTempus using the script integration API to perform actions like getting and setting Job Variable values, setting the Checkpoint for the job, logging messages to the Job Log, and sending e-mail messages.

Kinds of Scripts

There are two broad categories of scripts in adTempus:

- **Task Scripts** are scripts that run as the tasks of Job Steps. Generally the purpose of a task script is to accomplish a task that is external to adTempus.
- **Extension Scripts** are scripts that run to augment or customize the behavior of adTempus.

Task Scripts

You may already have scripts on your computer—such as VBScript (vbs) or PowerShell (ps1) scripts—that you use to perform various processing tasks. You can execute these scripts in adTempus using the Script Execution Task.

We also include Windows batch files in this category, because they are in some ways treated like other scripting languages within adTempus . However, since they are in other ways quite different from other scripting platforms, batch files are run using the Program Execution Task instead of the Script Execution Task.

The following scripting languages are supported for Task Scripts:

- .NET scripting using VB.NET and C#. This exposes the full power of the Microsoft.NET programming framework, without the need for a separate development environment or compiler.
- Windows Script Host (WSH) scripts such as VBScript and JScript.
- Windows PowerShell scripts.
- Windows batch files (run using the Program Execution Task).

Extension Scripts

adTempus lets you use scripts to extend and customize the behavior of adTempus. Scripts can be used:

- As Conditions, to determine whether a job or job step should run.
- As Actions, to perform quick tasks within adTempus.
- In Response Events, to determine when Responses should run.
- To determine whether a job step executed successfully.

- To dynamically determine the command-line parameters for a program to be executed by adTempus.
- To define inline functions, which allow you to call scripts and insert their return values anywhere that Job Variables are supported.

The following scripting languages are supported for Task Scripts:

- .NET scripting using VB.NET and C#. This exposes the full power of the Microsoft.NET programming framework, without the need for a separate development environment or compiler.
- Windows Script Host (WSH) scripts such as VBScript and JScript.

Windows Script Host scripts are supported primarily for backward compatibility. When creating new scripts you are encouraged to use VB.NET or C#, which provide a much more powerful programming environment.

In certain contexts within adTempus the WSH script languages are not supported, and will not appear in the list of available languages. In other contexts scripts written in these languages may not have access to all of the same adTempus context information or features as a .NET script.

Shared Scripts

Shared Scripts are scripts that can be used by more than one job. Any changes made to a Shared Script affect all jobs that use the script.

Shared Scripts are managed through the Shared Scripts view in the adTempus Console. When you create a Shared Script, you can use the security settings to determine which users are allowed to use or modify the script.

Script Libraries

Script Libraries allow you to create libraries of commonly-used code or data that can be shared among scripts in adTempus. This is equivalent to creating a "function library" or "class library" to be used by adTempus scripts.

Script Libraries are managed through the Script Libraries view in the adTempus Console. When you create a Script Library, you can use the security settings to determine which users are allowed to use or modify the script.

Related Concepts

Script 314

Related Topics

Script Properties Window 316
Interacting with adTempus from Scripts 326

Returning a Result From a Script 329
Inline Functions 323
Script Execution Task 199
Script Library 320

Installation

Installing

This topic discusses adTempus installation options and the installation process.

We recommend that you view the online version of this topic to ensure you have the latest information.

If you are upgrading from a previous version of adTempus, please review the upgrade information.

Your license information can be entered at the end of the setup process, or after setup using the License Manager tool in the adTempus program group on the Windows Start menu.

adTempus Components

adTempus is a "client/server" application, divided into two primary components:

- The adTempus Service (or "engine") does all the work of running jobs. The adTempus service must be installed on each computer on which jobs should execute. Throughout this document we refer to the Service, and the computer on which it is running, as the "server."
- The adTempus Console is the user interface for adTempus. It is used to configure and monitor adTempus. The Console may run on the same computer as the Service, or from a remote computer. A single Console can be used to monitor any number of adTempus services. Throughout this document we refer to the Console, and the computer on which it is running, as the "client."

If you are installing for the first time, you should perform a complete installation on the computer on which you want to run scheduled jobs. Optionally, you may install the client components on additional computers if you wish to administer adTempus remotely.

The remainder of this topic assumes you are performing a full installation of the software (service and Console). If you only need to install the Console (to perform remote administration), see the Console Installation topic.

You must purchase a license for each computer on which you install the server components (the adTempus Service). You may install the Console on any number of additional computers without purchasing additional licenses. See the licensing overview for more information.

Installation Process

The adTempus installation files are distributed in a self-extracting executable. When you run it, it will extract the setup files and then launch the installation manager (setup.exe).

The setup source files are not automatically deleted after setup completes. You can safely delete the files when installation is complete.

The installation manager executes separate phases of installation:

- **Software Installation.** Installs or upgrades the adTempus software.
- **License Activation.** Allows you to enter license information.
- **Database Installation and Configuration.** Installs SQL Server Express if required, and creates or upgrades your adTempus database.

Software Installation

The Software Installation phase is a standard Windows Installer setup package that will install any prerequisites prerequisites required for adTempus and then install or upgrade adTempus.

If a previous version of adTempus is already installed, you will have the choice to upgrade the current installation, or install the new version to a different location for a side-by-side installation.

Data Collection

Setup will ask your permission to enable anonymous feature usage reporting for adTempus. If you agree to this option, adTempus will collect and transmit to Arcana Development anonymous information about which features you use in the software and how often. The information gathered does not contain any data from your adTempus database or any other identifiable or potentially confidential information. You can enable or disable usage reporting later in the adTempus Console; separate options apply to the adTempus Server and to the Console itself.

Setup Type

Setup will ask you to select one of the following setup types:

- **Complete** installs all software components. This discussion assumes you are performing a complete (full) installation of the software.
- **Console Only** installs only the adTempus Console and other client utilities, which you can use to remotely administer the adTempus server installed on another computer.
- **Custom** allows you to select which components to install. Use this option if you want to install the server components without the client components, or to not install the reporting components.

Engine Mode

Setup will ask you to select the Engine Mode for the adTempus server.

If you are not planning to use the Distributed Scheduling features of adTempus, select the **Standalone** option. Otherwise, select **Master** if this instance will manage other adTempus instances, or one of the two Agent options if this instance will be managed by another instance. See the Distributed Scheduling Setup and Engine Mode topics for more information.

Note: The mode can be changed after installation using the Engine Mode program. There is no need to reinstall to change the engine mode.

Installation Wizard Completed

When you reach the end of the setup wizard, the adTempus software components are installed, but the adTempus database must be configured before adTempus can be used.

When you click Finish to close the setup wizard, the **License Manager** and **Database Installation and Configuration** tools will run.

In some cases (if a reboot was required during installation) the License Manager and Database Configuration Wizard tools may not start automatically after you click **Finish** in the installation wizard. If this happens, rerun setup.exe. The Installation Manager should skip software installation and move on to license configuration (if required) and database configuration.

License Configuration

Follow the instructions in the License Manager to activate adTempus if you have a license for this computer, or select evaluation mode if you do not.

You can review or add licenses later by running the **License Manager** tool from the adTempus group on the Start menu.

The License Manager may not appear if you already have a valid license for adTempus on this computer.

If you are upgrading from an earlier version of adTempus and have a license for the new version, the License Manager will help you upgrade your license.

Database Installation and Configuration

The Database Installation and Configuration tool will perform these actions:

- Install or upgrade SQL Server Express if appropriate
- Create or upgrade the adTempus database

See the Database Installation and Configuration topic for complete information on this process.

You don't need to be a database expert to install and configure adTempus: you can simply accept all the default settings in the database wizard and adTempus will install, configure, and manage the database for you.

If the database installation and configuration process fails, you *should not* uninstall/reinstall the adTempus software. After determining and resolving the cause of the failure, you can rerun the **Database Configuration Wizard** from the adTempus group on the Start menu.

Engine Mode

Each adTempus installation can operate in one of four **Engine Modes**: Standalone, Master, Agent, or Agent + Standalone.

If you are not using the Distributed Scheduling features of adTempus, you will use **Standalone** mode. If you are using Distributed Scheduling, you will have one **Master** computer and one or more **Agent** computers.

The Engine Mode is selected during installation but can also be changed after installing using the Engine Mode utility.

Standalone

In Standalone mode, Distributed Scheduling is not used. Distributed Scheduling options are not available in the user interface.

However, you can still create links between jobs on multiple computers using the Job Control Action , Job Condition, and Job Trigger.

Master

In Master mode, adTempus can run jobs on the host computer and on remote computers where adTempus is running in Agent mode.

Agent + Standalone

In Agent + Standalone mode, adTempus can run jobs sent to it by a Master computer and can also be used just like a standalone adTempus installation: you can connect to it with the Console and define jobs that are managed locally and not controlled by the Master.

When you connect to an Agent using the Console, you can view distributed jobs, but you cannot modify them. All changes to these jobs must be made on the Master.

An Agent can only be the Agent for one Master. Once a Master has connected to an Agent, that Master owns the Agent, and no other Master can connect to it.

An adTempus that uses this mode requires a full adTempus license, not an agent license.

Agent Only

In Agent Only mode, all jobs are managed by the Master. You cannot define local jobs.

An Agent can only be the Agent for one Master. Once a Master has connected to an Agent, that Master owns the Agent, and no other Master can connect to it.

ADTEMPUS

System Requirements and Prerequisites

System Requirements

adTempus runs on Windows platforms only. The following operating systems are supported:

- Desktop Operating Systems: Windows 7, Windows 8, or later.
- Server Operating Systems: Windows Server 2008 (not supported on Server Core Role), Windows Server 2008 R2 (not supported on Server Core Role), Windows Server 2012, or later

adTempus is available in a 32-bit version and a 64-bit version (x64 only; Itanium is not supported). Be sure you are running the correct installer for your platform: the installer name will have "win32" or "win64" in the name.

For Windows Server 2003 (or if you plan to run the adTempus server on Windows XP), see additional considerations before installing.

Disk Space Requirements

The adTempus software requires approximately 800MB of disk space.

The size of the adTempus database will vary depending on the amount of data in adTempus. An average installation should expect 1 to 4 GB of database space.

The SQL Server database engine, if used, requires approximately 1GB of space. In addition, SQL Server setup may require up to 6GB of free space on the System drive during installation (even if the software is being installed to a different drive).

Prerequisites

The following software components are required by adTempus, and will be installed if they are not present:

- Microsoft .NET Framework, version 4.0

Database Requirements

By default, adTempus will install, manage, and use its own instance of Microsoft SQL Server 2012 Express. If you prefer to use your own existing database server, see the Database Selection topic for additional information.

Installing on Windows Server 2003

Windows Server 2003 is supported by adTempus but the installation requirements are different than for newer platforms. Please review the following information before installing on Windows Server 2003.

Use the Correct Setup Package

The setup package you downloaded from our Web site should have "ws2003" in its name (e.g., "adtempus400_win32_ws2003.exe"). If it does not, return to the download page to download the correct package, which includes SQL Server Express 2008 instead of SQL Server Express 2012 (see below).

Windows Installer 4.5 Required

Windows Installer version 4.5 or later is required for adTempus setup but is not included in the adTempus setup package. To determine your current version of Windows Installer, run the following command from a command prompt to display the version number:

```
msiexec /?
```

If you do not have version 4.5 or later, download and install it before running adTempus setup.

Installation of Windows Installer will require a computer restart.

Additional Prerequisites

adTempus will download and install some additional prerequisites if necessary, including Microsoft .NET Framework version 4.0. This installation may require a restart.

SQL Server 2008 Express Used Instead of SQL Server 2012

The standard distribution of adTempus 4 uses SQL Server 2012 Express as the database engine. However, SQL Server 2012 does not support Windows Server 2003. The adTempus setup package for Windows Server 2003 therefore includes SQL Server 2008 Express instead.

Upgrading from Previous Versions

This topic describes the process for upgrading an adTempus installation from an earlier version.

We recommend that you view the online version of this topic to ensure you have the latest information.

If you are upgrading from a previous adTempus version, you can (and should) install version 4 without removing the previous version. The setup process will upgrade your existing installation.

Please review the information below for changes that may cause problems for upgrading users. See the What's New topic for an overview of new features.

If you are coming from version 2, please also see the Upgrading from adTempus 2.x topic.

Side-by-Side Installation

Beginning with version 4, adTempus supports side-by-side installations of the server software, meaning that you can install adTempus 4 on a computer alongside a version 3 installation, and run both versions of the software. See Side-by-Side Installation for more information.

Database Changes

This upgrade makes major changes to the adTempus database. Once it has been upgraded, the database cannot be used with previous versions of adTempus. Rolling back an upgrade therefore requires rolling back the database to a backup (the database configuration tool will create a backup before it makes changes).

If you are using the MSDE or SQL Server 2005 Express database engine provided with earlier versions, setup will upgrade the database engine to SQL Server 2012 Express (SQL Server 2008 Express on Windows Server 2003 platforms). This upgrade is required if you are still using MSDE, and strongly recommended for SQL Server 2005 Express.

If you have reports or programs that read the adTempus database directly, they will require modification to work with the new schema. Relations between tables are now modeled in the database, which will make it easier to understand the table structure.

adTempus Service Account Requirements

The adTempus service must run under the Local System account for full functionality. Running the service under a user account is no longer fully supported. This is necessary because there are some actions that adTempus needs to perform that Windows does not allow if the service is running under a user account. If you were previously running adTempus under a user account to allow access to a remote SQL Server database, you should reconfigure to either use SQL Server security or grant the adTempus machine account access on the remote database server. If you must use a user account for the adTempus service, review requirements and limitations here.

Upgrading Multiple Servers

adTempus instances from two different versions cannot communicate with each other. If you are using Distributed Scheduling, you must upgrade the Master and all Agents at the same time.

Linked Servers

If you have servers that are linked together through Job Control Actions or Job Conditions, you must upgrade those servers at the same time for those features to continue working. You will also need to establish trust relationships between the servers using the Linked Servers window. adTempus will log an Alert with ID 5270 at service startup if it finds remote targets for which no Linked Server definition has been created. You will need to configure the trust relationship before the conditions or responses can work.

Security Changes

The adTempus security model has changed, and will require review and adjustment of your security configuration after the upgrade. See the Security Changes topic for more information.

Additional Breaking Changes

adTempus Console does not work across major versions

The adTempus Console for version 4 cannot connect to version 3 servers or vice versa. We hope to support backward-compatibility in future major releases.

If you will need to continue administering servers running adTempus 3.x, you can still install the 4.x Console on your computer and use both versions of the Console concurrently.

Note that you can install the adTempus 4.0 Console on a computer with version 3.x or 2.x installed, but you cannot install the version 3.x or 2.x Console on a computer after you have installed version 4.0. Therefore if you need to be able to run both versions of the Console on a computer, be sure to install the 3.x or 2.x version first and then the 4.0 version.

The adTempus export/import format has changed

adTempus 4 uses a slightly different format for the export/import file, but is able to read and write the version 3 file format.

WMI trigger selection script changes

If you are using scripts to filter events for WMI Triggers, changes are required. Filter scripts must now be .NET scripts (VB.NET or C#); WSH scripts (VBScript) are not supported.

Job Log message codes for Scripts

If you have scripts that log messages to the Job Log using the adTempus.LogMessage method, the handling of the messageID has changed: the value that you pass to the LogMessage method will be increased by 100,000 before being logged to the database, to avoid overlap with system-defined message codes. That is, if you call LogMessage with a message ID of 42, your message will appear in the Job Log with a message ID of 100042.

All messages present in the database at the time of the upgrade will be converted to the new values.

If you are querying the adTempus database for specific message IDs from your scripts, you will need to modify your queries accordingly.

No more override of held jobs

The way that jobs/groups/queues are held/released has been changed. Along with this the option to run a job even if it is held (previously available in the execution options for a Job Control Action and when executing a job on demand) has been removed: it is not possible to override the job's Held status. If you have Job Control Actions that rely on this option you will need to change your configuration.

User interaction setting changes

The user interaction settings for jobs have been simplified, and the option to run jobs in the "console" session has been removed because it is less relevant/useful since session isolation was introduced in Windows Server 2008.

The adTempus API has changed

The adTempus API has changed for version 4. Please see the API documentation for more information.

New Web client

Due to the API changes discussed above, the Web-based monitoring client available with version 3 will not work with adTempus 4. A new Web Console is available for version 4.

Upgrading from adTempus 2.x

If you are upgrading from adTempus 2.x please review the changes between version 2 and 3 from the adTempus 3 user guide.

Distributed Scheduling Configuration

If you are using the Distributed Scheduling (Master/Agent) features in adTempus 2.x, your assignment of Jobs to Agents **will not** be preserved if you upgrade directly to adTemus 4 or later. You will need to configure Queues and assign jobs to them manually after the upgrade completes.

If you would prefer an automated conversion, you can upgrade your installation to adTempus 3 first (which will perform an automated migration of Distributed Scheduling settings from jobs to queues) and then upgrade again from adTempus 3 to the current version.

If you are not using Distributed Scheduling, this upgrade limitation does not apply to you.

Security Changes from Version 2

adTempus 3 introduces security inheritance to the security model. Users upgrading from previous versions of adTempus may have permissions individually assigned to jobs and other objects due to the limitations of the earlier security model. For ease of administration, users will in most cases want to remove object-level permissions in favor of inherited permissions. For example, if all your jobs have the same security settings, earlier versions of adTempus

★required you to make those settings on each job. With security inheritance, you can make those settings at the root job level and have all jobs inherit them.

Therefore you may need to remove the existing explicit permissions from the individual objects, so that they can inherit all of their security settings. Rather than edit each object individually, you can modify a higher level of the security hierarchy and tell adTempus to remove the non-inherited permissions on each lower level.

For example, if you want all jobs to have the same security settings, edit the settings for the root group (the "Jobs" node in the server tree). After setting the permissions you want your jobs to inherit, check the "Remove explicit permissions on all child objects" box. This will remove all permissions that have been set on individual groups and jobs, forcing them to use only the settings you have defined at the root level. If you now need to tailor the security of certain group and jobs, you can then edit them to override the inherited security.

Side-by-Side Installation

This feature is new for adTempus version 4.

Beginning with version 4, you can install the adTempus server software on a computer that is running a previous version of adTempus without removing or upgrading the old version. You will then have two separate adTempus services, operating completely independently.

You can also have more than one instance of adTempus 4 or later, using the Multiple Instance feature.

Installation Process

If the setup program detects an earlier version of adTempus already on the computer, it will ask you whether it should upgrade the existing version, or install the new version to a different location for a side-by-side installation.

When you choose the side-by-side option, you can set the Destination Folder to the same directory where the existing version is installed (e.g., "C:\Program Files\Arcana Development\adTempus"). The installer will install the adTempus software in a version-specific folder under the folder you choose.

The adTempus service will be named `adTempus$default`. If you have any scripts or tools that monitor or control the adTempus service, they will be pointed at the old version and will need to be changed to operate on the service from the new version if appropriate.

When the Database Configuration Wizard runs, be sure to choose the option to create a new database rather than using the existing database. Each installation of adTempus must have its own database.

Side-by-Side Installation with Version 2 or 3

If the previous version of adTempus on the computer is version 2 or version 3, some additional configuration is necessary, because the two adTempus services cannot both use the same communications port, which is 3760 by default. You will need to change the configuration for one of the instances so that it uses a different port number.

Changing the port number for an instance also means that all Consoles and peer servers connecting to it must use the new port number when connecting. You should therefore avoid changing the port number for the older instance until you have determined the related configuration changes that need to be made to match the new port number.

Licensing

Licensing Overview

This topic provides an overview of licensing requirements and procedures for adTempus. See the License Agreement for the complete agreement that governs your use of the software.

License Requirements

adTempus is licenses based on the "server" (the computer running the adTempus service). One license is required for each computer on which the service is being run. No additional licenses are required for computers running only the Console or other client tools.

adTempus is not licensed based on the number of processors on the computer, or based on the processor architecture (32-bit or 64-bit).

If you are running more than one instance of adTempus on a single computer, only a single license is generally required.

To determine the license status of a server, connect to that server in the Console, then select **About** from the **Help** menu.

Distributed Scheduling Agent Licenses

When you use the Distributed Scheduling features of adTempus, you install the adTempus server software on the Master computer and on each Agent computer. Each of these computers requires its own adTempus license. For example, if you have a Master computer and two Agent computers, you need three adTempus licenses.

Two license options are available for Agent computers:

- **A full adTempus license.** An Agent with a full adTempus license can run jobs sent to it by the Master computer. However, it can also run jobs that are managed locally.
- **A limited, agent-only license.** An Agent with an agent-only license can only run jobs sent to it by the Master computer. No jobs can be created or managed locally.

Evaluation Mode

If adTempus cannot find a valid license, it will run in "evaluation mode" for 30 days from the date when adTempus was first installed. During this time, adTempus is fully functional, so you can use all features. After 30 days adTempus will cease to run.

Entering License Information

To view or manage licenses for an adTempus server, run the "License Management" tool found in the adTempus program group on the Windows Start menu.

If you have lost your license information you can locate your license information using the online license management system at www.arcanadev.com/licensing.

License Agreement

This license agreement ("AGREEMENT") is a legal agreement between you (either an individual or a single entity) and Arcana Development, LLC ("Arcana Development") for the software product identified above ("SOFTWARE PRODUCT"), which includes the computer software and any associated media, printed materials, and "online" or electronic documentation ("PRODUCT COMPONENTS") distributed together under the product name shown above. The SOFTWARE PRODUCT also includes any updates and supplements to the original SOFTWARE PRODUCT provided to you by Arcana Development. Any software provided along with the SOFTWARE PRODUCT that is associated with a separate end-user license agreement is licensed to you under the terms of that license agreement. By purchasing, installing, copying, downloading, executing, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this AGREEMENT. If you do not agree to the terms of this AGREEMENT, do not install or use the SOFTWARE PRODUCT.

This AGREEMENT shall be governed by the laws of the State of Virginia and, in respect of any dispute which may arise hereunder, you consent to the jurisdiction of the federal and state courts sitting in Virginia.

Software Product License

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

The SOFTWARE PRODUCT contains some or all of the following types of software: "Server Software" that provides the core services or functionality of the SOFTWARE PRODUCT (the computer running the Server Software shall be referred to as the "Server"); and "Client Software" that allows other computers to gain access to or utilize the services or functionality provided by the Server Software (each computer running the Client Software shall be referred to as a "Client").

1. **Grant Of License.** This AGREEMENT grants you the following rights provided that you comply with all terms and conditions of this AGREEMENT:

   a. *"Licensed" Software.* If you have purchased or otherwise legally obtained a license to use the SOFTWARE PRODUCT (i.e., obtained from Arcana Development or its authorized agents a printed or electronic license certificate containing a valid license number or serial number for, and authorizing you to RUN, the SOFTWARE PRODUCT) the

SOFTWARE PRODUCT is considered “Licensed Software” (and you are considered a “LICENSEE”) and you are granted the following rights:

i. *“Server” Installation.* You may install, use, display, run, or otherwise interact with (“RUN”) the Server Software on a single computer (“Server”).

ii. *Multi-Computer Licenses*. If you have obtained a multi-computer license you may RUN additional copies of the Server Software up to the number of copies specified in your license certificate.

iii. *Site Licenses*. If you have obtained a site license you may RUN as many copies of the Server Software as you wish, on computers owned by you or under your sole control, at each physical location (“site”) for which you have obtained a license. Your site license certificate specifies the site for which the license is valid; the license does not entitle you to RUN the Server Software on computers that are not at the designated site.

iv. *Unlimited Use License.* If you have obtained an unlimited use license you may RUN as many copies of the Server Software as you wish, on computers owned by you or under your sole control, regardless of location.

v. *“Client” Installation.* Regardless of the kind of license you have obtained, you may RUN the Client Software on as many computers in as many locations as you wish for the purpose of interacting with the licensed Server Software.

vi. *Backup Server Installation*. If the Server Software is installed on a computer for which you maintain a passive fail-over or backup server, you may RUN the Server Software on a temporary basis on that backup server, for backup support only, without obtaining an additional license.

b. *“Evaluation” Software.* If the SOFTWARE PRODUCT is labeled “Evaluation” or “Trial” or was provided to you by Arcana Development or its agents as evaluation software, and/or if you have not obtained a license for the SOFTWARE PRODUCT as described in 1.a above, the SOFTWARE PRODUCT is considered “EVALUATION SOFTWARE” (and you are considered an “EVALUATION USER”) and you may RUN the SOFTWARE PRODUCT on any number of computers for up to 60 days for demonstration, test, or evaluation purposes. Your evaluation period commences at the time you first install a copy of the SOFTWARE PRODUCT; installing the SOFTWARE PRODUCT on a different computer or reinstalling the SOFTWARE PRODUCT on the same computer does not extend this evaluation period. At the end of the evaluation period you must either legally obtain a license to the SOFTWARE PRODUCT or uninstall and destroy all copies of the SOFTWARE PRODUCT in your possession.

c. *“Bundled” Software.* If the SOFTWARE PRODUCT was provided to you (by Arcana Development or one of its licensees) with or as part of another software application (the “CONTAINING APPLICATION”), the SOFTWARE PRODUCT is considered “BUNDLED SOFTWARE” and you may install the SOFTWARE PRODUCT only as part of the CONTAINING APPLICATION. You are licensed to use the SOFTWARE PRODUCT only as long as you are licensed to use the CONTAINING APPLICATION.

d. *Reservation of Rights.* All rights not expressly granted are reserved by Arcana Development.

2. Description of Other Rights and Limitations.

a. Version Limitation. Your license certificate contains a specific version number. This AGREEMENT permits you to install the same or a lower version of the SOFTWARE PRODUCT as the version number specified on the license certificate. If you have obtained this SOFTWARE PRODUCT as an upgrade to a previous version for which you have a license you are permitted to RUN the SOFTWARE PRODUCT even though it may have a later version number than is specified on your license certificate, provided you meet all eligibility requirements for the upgrade as described in section 3.

b. *Limitations on Reverse Engineering, Decompilation, and Disassembly.* You may not reverse engineer, decompile, disassemble, or modify the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

c. *Separation of Components.* The SOFTWARE PRODUCT is licensed as a single product. Its component parts may not be separated for use on more than one computer except for the separation of Server Software and Client Software as provided in this AGREEMENT.

d. *Rental.* You may not rent, lease, or lend the SOFTWARE PRODUCT.

e. *Support Services.* Arcana Development may provide you with support services related to the SOFTWARE PRODUCT ("SUPPORT SERVICES"). Use of SUPPORT SERVICES is governed by the Arcana Development policies and programs described in the user manual, in "online" documentation, and/or in other materials provided by Arcana Development. Any supplemental software code provided to you as part of the SUPPORT SERVICES shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this AGREEMENT. With respect to technical information you provide to Arcana Development as part of the SUPPORT SERVICES, Arcana Development may use such information for its business purposes, including for product support and development. Arcana Development will not utilize such technical information in a form that personally identifies you.

f. *Software Transfer.* The initial LICENSEE of the SOFTWARE PRODUCT may make a one-time permanent transfer of this AGREEMENT and SOFTWARE PRODUCT only directly to an end user. This transfer must include all of the SOFTWARE PRODUCT (including all component parts, the media and printed materials, any upgrades, this AGREEMENT, and, if applicable, the printed and/or electronic license certificate). Such transfer may not be by way of consignment or any other indirect transfer. The transferee of such one-time transfer must agree to comply with the terms of this AGREEMENT, including the obligation not to further transfer this AGREEMENT and SOFTWARE PRODUCT. Site licenses and unlimited use licenses may not be transferred without the consent of Arcana Development. BUNDLED SOFTWARE may not be transferred except as part of a transfer of the CONTAINING APPLICATION, if such a transfer is allowed under your license for that software.

g. *Termination.* Without prejudice to any other rights, Arcana Development may terminate this AGREEMENT if you fail to comply with the terms and conditions of this AGREEMENT. In such event, you must destroy all copies of the SOFTWARE PRODUCT and all of its component parts.

3. **Upgrades.** If the SOFTWARE PRODUCT is labeled as an upgrade, you must be properly licensed to use a product identified by Arcana Development as being eligible for the upgrade in order to use the SOFTWARE PRODUCT. A SOFTWARE PRODUCT labeled as an upgrade replaces and/or supplements (and may disable) the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this AGREEMENT. If the SOFTWARE PRODUCT is an upgrade of a component of a package of software programs that you licensed as a single product, the SOFTWARE PRODUCT may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

4. **Copyright.** All title and copyrights in and to the SOFTWARE PRODUCT (including but not limited to any images, photographs, animations, video, audio, music, and text incorporated into the SOFTWARE PRODUCT), the accompanying printed materials, and any copies of the SOFTWARE PRODUCT are owned by Arcana Development or its suppliers. If this SOFTWARE PRODUCT contains documentation that is provided only in electronic form, you may print as many copies as you wish of such electronic documentation, provided that it is for your use only. You may not copy the printed materials accompanying the SOFTWARE PRODUCT.

5. **Copies.** You may make as many copies as you wish of the installation media (if provided) or downloaded installation package, for the purpose of installing the SOFTWARE PRODUCT on computers for which you have obtained a license. After installation of the SOFTWARE PRODUCT pursuant to this AGREEMENT, you may keep the original media on which the SOFTWARE PRODUCT was provided by Arcana Development for backup or archival purposes. Except as expressly provided in this AGREEMENT, you may not otherwise make copies of the SOFTWARE PRODUCT or the printed materials accompanying the SOFTWARE PRODUCT.

6. Redistribution of Software. Copies of the SOFTWARE PRODUCT (including evaluation copies) may not be given, sold, or otherwise transferred to others without the prior permission of Arcana Development (except in the case of a license transfer as provided in section 2.f above). Software download web sites and other similar distribution services are permitted to redistribute evaluation copies provided that an evaluation copy or listing request was submitted to the service by an authorized agent of Arcana Development.

7. **U.S. Government License Rights.** This SOFTWARE PRODUCT is provided to the U.S. Government with the commercial rights and restrictions described elsewhere herein.

8. **Export Restrictions.** You may not export or re-export the SOFTWARE PRODUCT or any part thereof to any country, person or entity except in compliance with U.S. export restrictions. You specifically agree not to export or re-export any of the SOFTWARE PRODUCT (i) to any country to which the U.S. has embargoed or restricted the export of goods or

services, which currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Sudan and Syria, or to any national of any such country, wherever located, who intends to transmit or transport the SOFTWARE PRODUCT back to such country; (ii) to any person or entity who you know or have reason to know will utilize the SOFTWARE PRODUCT in the design, development or production of nuclear, chemical or biological weapons; or (iii) to any person or entity who has been prohibited from participating in U.S. export transactions by any federal agency of the U.S. government. You warrant and represent that neither the BXA nor any other U.S. federal agency has suspended, revoked or denied your export privileges.

9. **Use in Hazardous Environments.** THIS SOFTWARE PRODUCT IS NOT DESIGNED, MANUFACTURED, OR INTENDED FOR USE OR RESALE IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF THE SOFTWARE PRODUCT COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

Disclaimer of Warranty

THIS SOFTWARE PRODUCT IS PROVIDED "AS IS." TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, ARCANA DEVELOPMENT AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT, WITH REGARD TO THE SOFTWARE PRODUCT, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.

BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH AND USES TO WHICH THE SOFTWARE PRODUCT MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY SOFTWARE PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE SOFTWARE PRODUCT.

LIMITATION OF LIABILITY. To the maximum extent permitted by applicable law, in no event shall Arcana Development or its suppliers be liable for any special, incidental, indirect, or consequential damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use the SOFTWARE PRODUCT or the provision of or failure to provide Support Services, even if Arcana Development has been advised of the possibility of such damages. In any case, Arcana Development's entire liability under any provision of this AGREEMENT shall be limited to the greater of the amount actually paid by you for the SOFTWARE PRODUCT or U.S.$5.00; provided, however, if you have entered into an Arcana Development Support Services Agreement, Arcana Development's entire liability regarding Support Services shall be governed by the terms of that agreement. Because some states and jurisdictions do not allow the exclusion or limitation of liability, the above limitation may not apply to you.

Database Installation and Configuration

Database Installation and Configuration

The Database Installation and Configuration wizard is run at the end of adTempus installation and can also be run directly by selecting the **Database Configuration Wizard** from the adTempus group on the Windows Start menu.

We recommend that you view the online version of this topic to ensure you have the latest information.

Overview

Each instance of the adTempus service (including Agent installations) requires its own database in which to store data.

By default adTempus will install and use a SQL Server 2012 Express database. This database is included at no cost to you and does not require you to do any database configuration or management. This is the recommended configuration for most users.

You may also choose to use your own SQL Server instance to host the adTempus database.

See the Database Selection topic for more information on choosing a database.

Even if an instance of SQL Server Express has already been installed on the computer by another application, you should allow adTempus to install its own instance so that the applications do not interfere with each other.

Configuration Process for New Installation

For a new adTempus installation, the Database Installation and Configuration wizard will be presented. The following database options are available:

- Install SQL Server Express. As discussed in the Database Selection topic, this is the recommended option.
- Use an existing SQL Server instance

Configuration Process for Upgrade

If you are upgrading from an earlier version of adTempus, the Database Wizard will allow you to review your current database settings and will upgrade your database if necessary.

If you are using an older version of SQL Server Express than what is included with the current version of adTempus, you will be given the option to upgrade it.

Database Selection

adTempus stores most of its data in a relational database. For the most part, you do not need to know anything about the database beyond making the initial determination about which database to use.

By default, adTempus will install the Microsoft SQL Server 2012 Express database engine, and this is the recommended configuration for most users. However, you may also choose to use your own existing database server.

In some cases users may exceed the capabilities of SQL Server Express in terms of data storage or database load. In such scenarios you will need to use a standalone SQL Server instance.

If you are using Distributed Scheduling, each adTempus Agent requires its own database. We recommend that you use SQL Server Express for the Agents even if you choose to use a full SQL Server instance for the Master. This eliminates the database server as a single point of failure in your adTempus configuration.

Using the Default Database Engine

The default database engine, Microsoft SQL Server 2012 Express, is included as part of adTempus and does not require an additional license purchase. When you choose to use this option, adTempus Setup will install, configure, and manage SQL Server Express for you: you will not need to interact directly with the database server. A backup of your adTempus data will be created each night.

Note that even if you already have SQL Server 2012 Express installed on the computer, adTempus will install a new SQL Server instance.

SQL Server 2012 Express requires approximately 270MB of disk space for the database program files.

This is the recommended database configuration because:

- It requires no database knowledge, configuration, or management by you, the adTempus user.
- It ensures that adTempus will always be able to connect to its database, because the database will be on the same computer as adTempus.
- It isolates adTempus from problems such as database performance issues or database server shut downs that may be caused by other applications or other business needs.

Using Your Own Database Server

If you already have a Microsoft SQL Server database server (version 2005 or later), you may choose to host the adTempus database on this server instead of using the SQL Server Express instance available with adTempus.

See the Using a Standalone SQL Server Instance topic for more information about using your own SQL Server instance.

As noted above, we recommend using the SQL Server Express instance because it provides isolation and reliability for adTempus. If you choose to use your own SQL Server instance, you should keep in mind that adTempus requires continuous access to the database server. If SQL Server is shut down, or the network connection between computers is lost, this will cause problems in adTempus and may lead to missed or failed jobs.

Related Concepts

Database Installation and Configuration 113

SQL Server Express

When you choose the SQL Server Express option, the database configuration tool will install SQL Server Express, then create the adTempus database.

If necessary you may change the location where SQL Server program files should be installed. These files should *not* be placed in the adTempus installation directory.

Leave the **Show SQL Server installation progress window** box checked to see detailed progress information displayed by the SQL Server setup program during installation. This is recommended, as the SQL Server installation process can be lengthy.

Click **Next** to continue to the **Select Actions** page. All actions will be selected on this page, which will be appropriate for most installations.

Click **Next**, then **Next** again to begin setup.

At this time, the setup program will install SQL Server Express, create and initialize the adTempus database, and start the adTempus service using the new database.

If this process fails for any reason, review the error messages for more information, and contact technical support. The adTempus software will remain installed, but you will not be able to start the adTempus service until the database is successfully installed and configured. Uninstalling and reinstalling the adTempus software will not resolve the problem.

Related Concepts

Database Installation and Configuration 113

Using a Standalone SQL Server Instance

adTempus can use an existing SQL Server database server to host its database. SQL Server may be running on the same computer as adTempus, or on a remote computer. SQL Server 2005 and later are supported.

Information in this topic refers to the Database Installation and Configuration program, which is the program that configures SQL Server and adTempus. This program is run either by the adTempus setup program, or manually, at a later time.

Installation Requirements

The Database Installation and Configuration program must be run with sufficient authority to create databases and users on the SQL Server. When you run the program, therefore, you must either be logged in under a Windows account that has system administrator authority for SQL Server, or have a system administrator user ID and password that the tool can use.

SQL Server must be running.

Security Setup

If SQL Server is on the same computer as adTempus, you can generally use Integrated (Windows) security, which means that adTempus can connect to the database without an explicit user ID and password. This is possible because the adTempus service runs under the SYSTEM account, which is a member of the Administrators group on the computer, which by default has system administration authority for SQL Server.

If you have changed SQL Server security settings such that members of the computer's Administrators group do not automatically have system administration authority in SQL Server, you will need to either make the necessary changes to allow the SYSTEM account full control over the adTempus database, or enable SQL Server security and have adTempus use that.

If SQL Server is on a different computer from adTempus, you must use one of the following approaches:

- Enable SQL Server security on the server and allow adTempus to use that instead
- Configure SQL Server to grant permission to the adTempus server's computer account

The Database Configuration Wizard will help you with the necessary settings for either option.

Hosting Multiple adTempus Instances

If you are installing adTempus on several servers, you can use a single SQL Server instance to host the databases for all of the adTempus instances. However, each instance of adTempus must have its own separate database on the server.

The Database Installation and Configuration program handles this automatically, by including the name of the adTempus server in the database name. For example, you have installed adTempus on three servers: ServerA, ServerB, and ServerC. You will use the SQL Server on ServerX to host the databases for all three adTempus installations. When you run the database migration tool, it will create the databases "adTempus_ServerA," "adTempus_ServerB," and "adTempus_ServerC."

Database Backups

By default adTempus does not back up the adTempus database when you are using a standalone SQL Server instance. You can configure adTempus to create a daily backup in the Server Options window if the database is not included in another backup policy.

Related Concepts

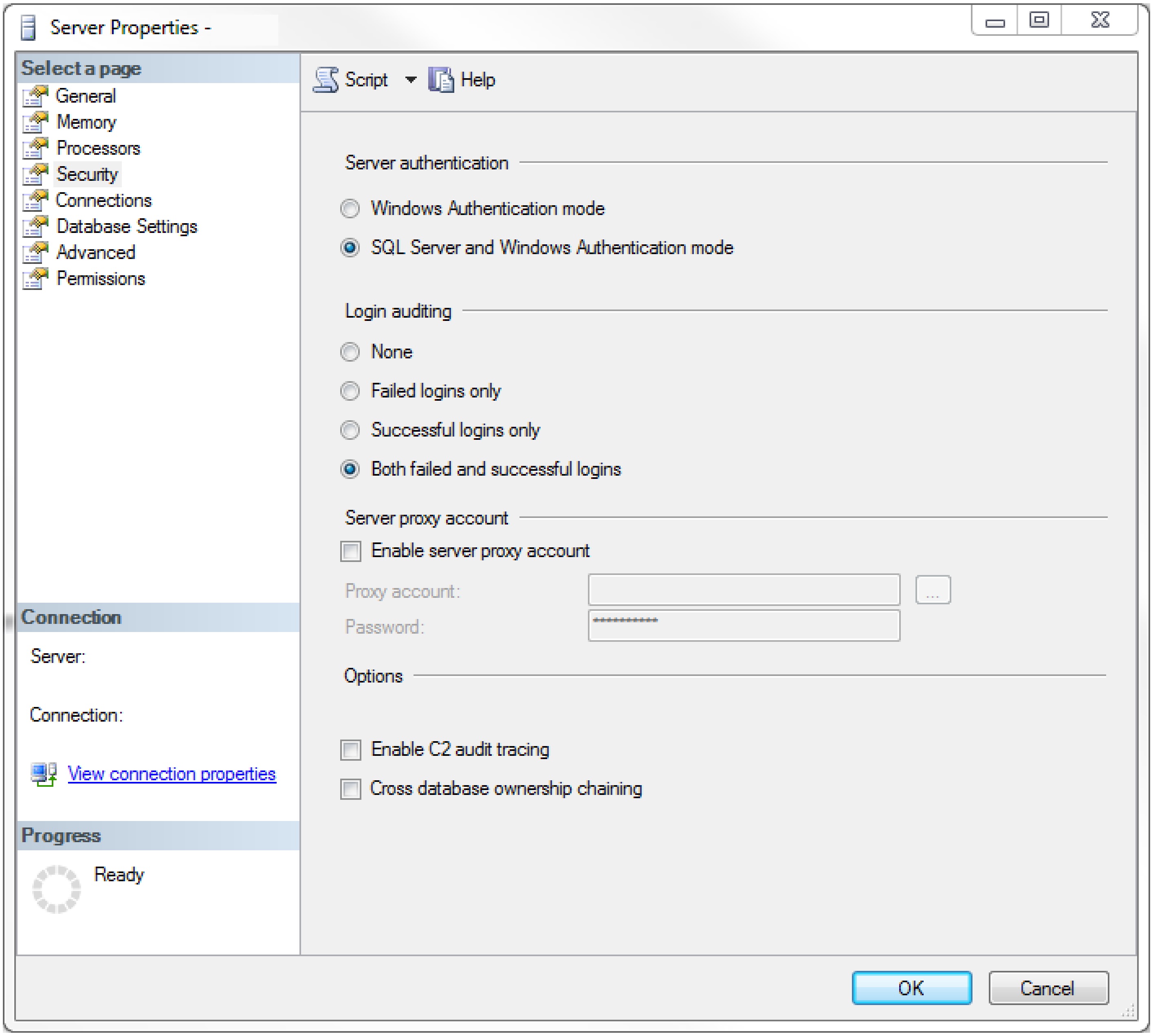
Database Installation and Configuration........113

Enabling SQL Server Security

To use SQL Server authentication (explicit user ID/password) to connect to the adTempus database, SQL Server authentication must be enabled for the database server.

Important: Before making this change, be sure that you have provided a strong password for the sa user account, and consider the security implications of this security mode. You may prefer instead to continue using Windows integrated security and grant the adTempus computer account access to the adTempus database.

To review or change the security setting, run SQL Server Management Studio and connect to the database server. Right-click the server name in Management Studio and select Properties, then go to the **Security** page:



Server:

Connection:

Ready

Server authentication

○ Windows Authentication mode

● SQL Server and Windows Authentication mode

Login auditing

○ None

○ Failed logins only

○ Successful logins only

● Both failed and successful logins

Server proxy account

□ Enable server proxy account

Proxy account:

Password: ***********

Options

□ Enable C2 audit tracing

□ Cross database ownership chaining

Select **SQL Server and Windows Authentication** mode to enable SQL Server authentication. You can then create a login for adTempus to use.

Related Concepts

Database Installation and Configuration 113

Database Configuration Wizard Reference

SQL Server Express Download

This page is displayed if the Database Configuration Wizard is run outside of setup, or if you have downloaded a version of the setup package that does not contain SQL Server Express.

If you plan to use a standalone SQL Server instance or an existing SQL Server Express installation, you can ignore this message and continue.

Otherwise you need to point the Wizard to the SQL Server Express setup package, which you can obtain by extracting the setup files from the adTempus setup package.

Related Concepts

Database Installation and Configuration 113

MSDE Detected

If you are still using the Microsoft SQL Server Desktop Engine (MSDE) version of SQL Server that was included with adTempus versions 1 and 2, the installation must be upgraded to use a newer version of SQL Server Express.

The installer for SQL Server Express is included in the adTempus setup package. If you are running the Database Configuration Wizard separately from setup, the installer may not be available. In this case you can run the setup package to extract the setup files, then point the Wizard to the SQL Server Express installer extracted from the package.

Related Concepts

Database Installation and Configuration 113

Database Upgrade

This page is displayed if your adTempus database is from a previous version of adTempus and needs to be upgraded to be compatible with the current version.

Once the database has been upgraded, it cannot be used with previous versions of adTempus.

Before the upgrade process begins, the Database Configuration Wizard can make a backup of the database. If the upgrade fails, the database will be rolled back to the backup automatically.

You should uncheck this option only if you already have a current backup of the database.

Related Concepts

Database Installation and Configuration 113

SQL Server Express Upgrade

adTempus 4 includes SQL Server 2012 Express (or SQL Server 2008 Express for Windows Server 2003 platforms). If you are currently using an older version of SQL Server Express, adTempus will prompt you to upgrade the SQL Server installation. We strongly recommend that you perform the upgrade.

Upgrading SQL Server Express for adTempus only affects the adTempus instance of SQL Server Express: it does not affect any other instances of SQL Server Express or SQL Server.

To upgrade SQL Server Express, adTempus will perform the following steps:

- Detach the adTempus database
- Uninstall the old version of SQL Server Express
- Install the new version of SQL Server Express
- Reattach the adTempus database

The database will then be upgraded to match the current version of adTempus.

Related Concepts

Database Installation and Configuration 113

SQL Server Location

Specify the SQL Server instance to connect to.

adTempus requires a continuous, reliable connection to the database server. If you choose to use a standalone database server, especially on a separate computer, you must ensure that it is always available for adTempus, and that adTempus is shut down any time SQL Server needs to be shut down.

SQL Server Location

Specify the local or remote location of the SQL Server that will host the adTempus database.

Use a named instance of SQL Server

If the SQL Server instance is not the default SQL Server instance, specify the instance name

Related Concepts

Database Installation and Configuration 113

Standalone SQL Server

When you choose the standalone SQL Server option, you will next be prompted for the information needed to connect to the database server and create the adTempus database.

Please review the Using a Standalone SQL Server Instance topic for information about SQL Server requirements and configuration.

The **Database Name** will default based on the name of the computer where adTempus is running. You may change the database name to something else if you prefer, but keep in mind:

- The database must **not** already exist, as the setup program will create the database.
- Each adTempus instance requires its own database, so if you will be hosting data for more than one adTempus instance on a single SQL Server, be sure the database names are unique.

Click **Next** to continue to the Authentication page, and specify SQL Server login credentials if necessary. These credentials are used only for the database creation and configuration process; adTempus itself will use different credentials as discussed in the Using a Standalone SQL Server Instance topic.

Click **Next** to continue to the **Select Actions** page. All actions will be selected on this page, which will be appropriate for most installations.

Click **Next**, then **Next** again to begin setup.

At this time, the setup program will connect to the database server, create and initialize the adTempus database, and start the adTempus service using the new database.

If this process fails for any reason, review the error messages for more information, and contact technical support. The adTempus software will remain installed, but you will not be able to start the adTempus service until the database is successfully installed and configured.

Related Concepts

Database Installation and Configuration.....113

SQL Server Authentication

Specify the credentials that the Database Configuration Wizard should use to connect to SQL Server.

The account you specify must have permission to create a new database on the server.

These credentials are used only for database configuration. You will specify the credentials for adTempus itself to use in a later step.

Related Concepts

Database Installation and Configuration.....113

Database Name

Specify a name for the adTempus database. The default name is `adTempus_`*`computername`*, where *computername* is the name of the computer where adTempus is being installed.

The database configuration tool will create this database; configuration will fail if the database already exists.

Each instance of adTempus (including Agent installations) must have its own database.

Related Concepts

Database Installation and Configuration....113

Database Already Configured

This page is displayed if the Database Configuration Wizard finds existing database configuration settings in the Registry.

In the pages that follow you can keep your existing settings, in which case the Wizard will upgrade your current database if necessary.

You may also change to use a different database, in which case the Wizard will created it and configure adTempus to use it.

Related Concepts

Database Installation and Configuration....113

Database Credentials

Specify the database credentials that adTempus will use to connect to the database. The account you specify must have read, write, and backup permissions for the database but does not need to be an administrator or database owner.

Database Credentials

Select the authentication type for the connection to SQL Server.

Windows Authentication

Choose Windows Authentication to use integrated security for the database connection. To use this option, the LocalSystem account on this computer must be granted the necessary login and database permissions on SQL Server.

In adTempus 3 and earlier, you could change the adTempus service to run under a user account instead of LocalSystem, and grant the necessary database access to that account. This configuration is no longer fully supported: adTempus must run under the LocalSystem account for all adTempus features to be available. More information.

If SQL Server is located on the same computer as adTempus, make the necessary security changes in SQL Server to grant permissions to the LocalSystem account.

If SQL Server is running on a different computer, additional configuration is required, and the SQL Server Computer Account page (see below) will be displayed when you click **Next**.

SQL Server Authentication

Choose SQL Server Authentication to connect using a user ID and password. To use this option, your SQL Server must be configured to allow SQL Server authentication.

If the account you want adTempus to use already exists, specify the user ID and password here.

Otherwise, adTempus can create the account for you, or you can copy the script shown at the bottom of the page and use it to create the account.

SQL Server Computer Account

If you are using Windows authentication for SQL Server, and SQL Server is running on a different computer than adTempus, you must create a machine login in SQL Server for the computer on which adTempus is running.

This is necessary because adTempus runs under the special LocalSystem account that is specific to the computer on which adTempus is running. Were you to grant permissions to "LocalSystem" in SQL Server, that would only grant permissions to the LocalSystem account on the database server itself, not to the LocalSystem account on the adTempus computer.

You may have adTempus create the necessary account for you, or you may review the script that adTempus generates and execute it yourself.

Related Concepts

Database Installation and Configuration........113

Select Actions

Select the actions the Database Configuration Wizard should perform (some actions may not be available depending on your scenario).

- **Install SQL Server Express.** The Database Configuration Wizard will install SQL Server Express.
- **Create the adTempus database.** The Database Configuration Wizard will create the adTempus database on the specified server.
- **Configure adTempus to use the database.** The Database Configuration Wizard will update the adTempus database settings in the Registry to point adTempus at the database.
- **Start the adTempus service.** The adTempus service will be started.

Related Concepts

Database Installation and Configuration 113

Administrator Provisioning

Specify the Windows users who should be designated as Administrators in adTempus.

If you are creating a new adTempus database or upgrading from an adTempus version earlier than 4.0, you must designate at least one Administrator so that you can log in to adTempus under that user account and perform adTempus configuration (such as creating additional users).

If your adTempus database was created by adTempus version 4.0 or later, you will already have administrators defined, and this step can be skipped.

See the Administrator Provisioning and Security Overview topics for more information.

Related Concepts

Database Installation and Configuration 113

Configuration Progress

Review your settings and click **Next** to carry out the actions you specified.

If you are installing or upgrading SQL Server Express, the SQL Server Express installer will run separately but will not require any input from you. After SQL Server installation completes, the remaining required actions will execute.

The Wizard will show each action that it takes. For a database upgrade, the Wizard will update its progress on each step of the conversion.

Successful Completion

If the database is successfully configured (and upgraded, if necessary) the Wizard will display a success message and, of you chose the option, start the adTempus service for you.

Database configuration is now complete. If this is a new adTempus installation, you should run the adTempus Console and proceed with initial configuration.

Failure

If the Wizard encounters an error while creating or upgrading the adTempus database, error information will be displayed.

If a database upgrade fails at a point from which the upgrade cannot resume, adTempus will roll back the database to the backup it took at the beginning of the upgrade process. In some cases, however, it will be possible to restart the conversion from the point of failure after problems are addressed. In this case, the database is not rolled back.

In most cases you will not be able to resolve problems reported by the Database Configuration Wizard if they relate to creating or upgrading the database. You should contact technical support for assistance. Be sure to include the exact error information that you received, and also the "adtdbconfig.log" file found in the "logs" directory under the adTempus program directory.

Do not uninstall and reinstall adTempus to try to resolve the problem if an error occurs during database configuration. This will almost never be useful, and in some cases may make things worse. The database configuration can be rerun after problems are addressed by running the Database Configuration Wizard from the adTempus group on the Start menu.

Related Concepts

Database Installation and Configuration 113

Uninstalling

adTempus can be uninstalled by using the **Programs and Features** (or **Add and Remove Programs**) tool in the Windows Control Panel.

You do not need to uninstall adTempus before you install a newer version: the setup program for the newer version will upgrade your existing installation.

License Deactivation

If you have activated the adTempus software on this computer using your license, the uninstall process will ask you if you want to "deactivate" the software and return the license to your license account. If you do not plan to reinstall the software on this computer, you should answer "Yes," to make the license available for use on another computer. If you will be reinstalling the software on this computer, you should not return your license.

Database Removal

Uninstalling adTempus will not remove the adTempus database engine (SQL Server 2012 Express), if installed. If you will be reinstalling adTempus on this computer, you should leave the database engine installed. If you will not be reinstalling adTempus, you should remove the database engine.

To remove SQL Server Express, select "Microsoft SQL Server 2012" in the Windows **Add or Remove Programs** tool and click **Remove**. In the Microsoft SQL Server 2012 Uninstall wizard, check the "Remove SQL Server 2012 instance components" box and select the " [[[Undefined variable Primary.SQL Express Instance Name]]]: Database Engine" instance, then click **Next** and continue with the removal process.

Be sure you select only the "[[[Undefined variable Primary.SQL Express Instance Name]]]: Database Engine" instance. If any other instances are listed, they were installed by other products and should not be removed.

Multiple adTempus Server Instances

This feature is new for adTempus version 4.

Beginning with version 4, you can run multiple instances of adTempus on the same computer. Each instance is a separate copy of the adTempus service, with its own database, settings, and other data, that runs independently of all other instances.

For example, if your main adTempus instance is running your production jobs, you could create a second instance for testing new versions of jobs. You can then use the Export tool to copy jobs from the test instance to the production instance.

If you need to have two different major versions of adTempus running on the same computer (e.g., 3.x and 4.x), see the Side-by-Side Installation topic.

Multiple instances differ from side-by-side installation because all of the instances use the same version of the adTempus software.

The first instance of adTempus that was installed by the setup program is called the "Default" instance.

What is an Instance?

Each adTempus instance runs as a separate copy of the adTempus service (each adTempus instance appears as a separate service in the Windows Service Manager). Because each instance runs as a separate service, it is isolated from other instances: you can stop or start each instance independently without affecting the other instances. Each instance also has its own database and other data.

While each instance runs as a separate service, all instances share the same installed version of the software. Therefore if you apply a software update, it will affect all instances.

Requirements for Multiple Instances

You can create as many instances on a single computer as your hardware can support. The practical limit will depend on memory and processor power.

In order to allow all instances to use the same port number for communications, you must enable and start the **Net.Tcp Port Sharing Service**, which is installed but disabled by default. To do this:

1. Open the Windows Services tool.
2. Locate the **Net.Tcp Port Sharing Service** and open its properties.

3. If the **Startup type** is not set to "Automatic," change it to "Automatic."

4. Apply the settings change, then start the service.

Managing Instances

To manage instances, run the **Instance Management** tool from the Start menu (**adTempus** 4**> adTempus Server Tools > Instance Management**) or run the "adtservertools" program from the adTempus program directory and select the **Instance Management** option.

Instances are not created or removed using the adTempus setup program. It is not necessary to rerun setup to manage instances.

Creating Additional Instances

To create a new instance, enter a name for the instance.The instance name can be any combination of letters, numbers, and underscores. Other characters are not allowed.

Generally you will want to use the default adTempus port. Beginning with version 4, adTempus instances can share the same port number so there is no need to use a different port for an instance.

Click **Create** to create the new instance. This creates a new adTempus service for the instance, named `adTempus$`*`instancename`* where *instancename* is the name you gave the new instance.

Once the instance is created, the tool will prompt you to run the database configuration wizard to create a database for the new instance. Each instance must have its own database.

Once the database has been created, you can start the new service and begin using the new instance (see Connecting to Instances below).

Removing Instances

To remove an instance, select it from the instances list and click **Delete**. The tool will stop the instance's service and delete the service. The database and registry settings will not be deleted, so you can re-create the instance later if necessary.

Connecting to Instances

To connect to an instance other than the Default, specify the instance name in the Server Connection Properties window when you create a new Server Connection in the Console.

Software Updates

If the adTempus server is connected to the Internet, it will check daily for adTempus software updates. If updates are available, a message will be logged in the Alerts view of the Console.

If the server is a Distributed Scheduling Master, software updates can be pushed to Remote Agents from the Remote Agents view.

If the adTempus Console is connected to a server that has a newer version of the adTempus software, the Console software can be updated by choosing the **Install Console update** command from the Help menu.

Console-Only Installation

The adTempus Console is the user interface component for adTempus. It is used to configure and monitor adTempus. The Console may run on the same computer as the Service, or from a remote computer. A single Console can be used to monitor any number of adTempus services. Throughout this document we refer to the Console, and the computer on which it is running, as the "client."

If you are installing the adTempus server component, you will install the Console at the same time.

If you need to install only the Console (so that you can connect to an adTempus server installed on another computer) you can perform a Console-only installation. This is done using the same setup program that is used to install the adTempus server components. When prompted to select the setup type, choose the **Console Only** option.

No license is required for Console-Only installations.

Getting Started

Getting Started

The following topics will help you get started using adTempus.

- If you haven't done so, you'll want to install the software.
- You should first read the Introduction to Key Concepts, will introduce you to the major "building blocks" you will be working with in adTempus.
- The adTempus Console is the administration tool for adTempus.
- If you have just installed adTempus, you will need to do some first-time setup.
- If you are upgrading from the Arcana Scheduler, you can import your Arcana Scheduler jobs.
- Finally you're ready to create a job.

Related Topics

Introduction 81
Key Concepts 81

First-Time Setup

If you have installed a new adTempus instance, there are a few initial settings that you will probably need to configure. You will need to be logged in to the computer under an account that was provisioned as an Administrator during installation.

To begin, run the adTempus Console from the adTempus group on the Windows Start menu. The Console should automatically connect to the local adTempus server.

Security Settings

During installation, one or more Windows users were added as adTempus users and provisioned as adTempus Administrators.

To allow other users to use adTempus, you must create adTempus logins for them and grant them the necessary permissions. See the Security Configuration Guidelines topic for more information.

Notification

If you plan to use e-mail or SMS notification (e.g., for sending messages about failed jobs), you must configure the notification options.

Holidays

Jobs can be configured to behave differently on holidays. If you plan to use this feature you should review the default holidays and modify the list as appropriate for your organization.

Shared Schedules

Shared Schedules allow you to use a single schedule to determine the days on which many jobs will run (each job still has its own settings for the time(s) at which it will run).

You may want to set up some initial shared schedules that are available to all users, and establish policies for their use. For example, you may want to define a "Weekdays" schedule for jobs that run Monday through Friday.

How To: Create a Basic Job

This example shows how to create a simple job to run the "Notepad" program from adTempus.

To create the job:

1. Select the Job Group that you want to create the job in. Right-click the group's name and select **New job** from the pop-up menu. (If you don't have any Groups defined, or want to create the job in the root group, right-click the **Jobs** node.)

2. The Job Properties window will open. Though adTempus offers many options, a simple job requires only a few things.

3. Enter a **Name**. Give the job a unique, descriptive name, such as "Run Notepad."

4. Enter the **User Account**. Type the user ID for the Windows user account the job should run under. If the account is a domain account, include the domain, for example `mydomain\myuserid`. When you leave the User Account box, you will be prompted to enter the password for this account. For this example, use your own user ID.

5. Click to the Steps page.

6. Click **Add...** and select the "Execute a program, batch file, etc." task type. This creates a Program Execution Task, which runs a program or batch file and is the most commonly-used kind of task.

7. The Program Execution Task properties window will open.

8. Next to the **Target** box, click the "**...**" button to browse for the file to execute. Browse to the Windows directory (usually "c:\windows") and select `notepad.exe`.

9. Click **OK** to save the task properties and return to the job properties.

10. Click **OK** to save the job.

11. The new job will now appear in the folder.

Because this job does not have any Triggers assigned to it, adTempus will run it only if you request it. To run the job:

1. Right-click the job's name and select the **Run** command. The Execute Job window will be shown.

2. Check the **Make the job visible on my desktop** option.

3. Click **OK** to start the job, then click **Close** to acknowledge the informational message.

4. The job's status in the Console will be updated to show that it is running, and the Notepad program will open on your desktop.

This example assumes you are running the adTempus Console on the same computer where the adTempus server is installed. If you are not, or if you are connected to that computer through a Remote Desktop or Terminal Services connection, you may not be able to see the Notepad program running on your desktop. In this case you will need to terminate the job by right-clicking it and selecting the **Terminate** command.

4. Close Notepad. The job's status will be updated to show that the job has finished.

Next Steps

Most jobs will have a Trigger so that adTempus will run them automatically. The most common trigger is the Schedule Trigger, which runs the job at specified days and times. We'll add a simple Schedule Trigger to the job you just created.

1. Double-click the job to display its properties, then click to the Triggers page.

2. Click the **Add...** button and select "Execute according to a schedule." The Execution Schedule Properties window will open.

3. On the **Date Selection** page, check the **Trigger every 1 days** option. These settings will cause the job to run every day.

4. Click to the **Time Selection** page.

5. Select the **Trigger at these times** option.

6. Below the list, type in a time when the job should run. For this example, enter a time that is two or three minutes in the future.

7. Click the **+** button to add the time to the list.

8. Click **OK** to save the schedule properties. The Schedule Trigger properties window now appears, where you could set additional trigger options.

9. Click **OK** to return to the job.

10. Click **OK** again to save the job and return to the Console.
11. After a moment you will see a **Next Start** time for your job.
12. When the scheduled time arrives, the job will start.

adTempus Objects

Job

Job

The Job is the central focus of adTempus. The job contains all of the information needed for adTempus to accomplish a specific goal for you. Each job contains one or more steps; each step executes a single task.

When you create or modify a job you are modifying a "template" for the job. Each time the job executes adTempus creates an instance of that job. Changes you make to a job do not have any effect on any instances that are already executing.

Information about each instance is retained in the job's history, which can be viewed from the Jobs View or the Execution History Query.

Security

The adTempus security framework determines who is able to manage a job within adTempus, but these settings do not affect what the job can do at run-time. Instead, for each job you must specify the Windows user account that the job will use. When adTempus runs the job it logs in the specified user and runs the job in that user's security context.

This means that any programs, scripts, or other tasks run by the job will have the same security limitations as if that user were running the program directly.

Related Concepts

Key Concepts 81

Related Topics

Steps Overview 145

Reference

Job Properties 133

Job Properties

The **Job Properties** window contains the settings for a Job.

The properties for the job are divided among the following pages:

General

Documentation

Variables

Recovery

Triggers

Conditions

Steps

Resources

Responses

Security

Related Concepts

adTempus Objects 133

Related Topics

Job General Page 134
Documentation Page 137
Variables Page 137
Job Recovery Page 137
Triggers Page 138
Conditions Page 140
Steps Page 140
Job Resources Page 141
Job Responses Page 142
Steps Overview 145

Job General Page

Location: Job Properties window

Name

Provide a descriptive name for the job (255 characters maximum). The name is used throughout adTempus to identify the job, and must be unique within the group that the job belongs to.

If you try to use a name that is already in use on another job, you will receive an error message when you try to save the job. Note that the other job with the same name may not be visible to you (based on security settings).

The name can be changed at any time without affecting links between jobs.

Job ID

The Job ID is a 38-character system-generated unique identifier for the job. This ID remains constant even if the name of the job changes. The Job ID is shown primarily for use in

⭐diagnostic situations but can be used instead of the name in command-line utilities such as adtExec.

Queue

Select the Queue to which this job is assigned.

Priority

Specify the priority of this job within the queue. In a queue that limits the number of jobs that can run simultaneously, jobs with higher priorities are executed before jobs with lower priorities. New jobs have a default priority of 100. A higher priority is indicated by a higher priority number: for example, a job with priority 200 has a higher priority (and will run before) a job with priority 100.

User Account

Specify the Windows user account that the job should be run under. See Entering User Credentials for more information on how to enter the credentials.

All tasks performed by this job will be run under the user account you specify. For example, when the job executes a program, the program will operate under the same security context (and restrictions) as if the specified user had run the job herself.

Run with highest privileges

If the specified **User Account** is an administrative account, check the **Run with highest privileges** option to have administrative privileges enabled when the program runs. This is equivalent to using the "Run as administrator" command in Windows. If this option is not checked, administrative privileges will not be enabled, and the program may not behave as you expect.

If the**User Account** is not an administrative account, checking this option has no effect on program execution.

This option is not available if the adTempus server is running Windows XP, Windows Server 2003, or earlier.

User Interaction

This option has been simplified from the settings available in adTempus 3.

Select whether a user needs to be able to see or interact with programs run by your job.

Each job runs in one of the following sessions:

- **Hidden.** (Default and recommended setting) The job will be run in a hidden session. If the programs run by the job display a user interface, no user will be able to see them.
- **Interactive if available.** If the user whose account is used to run the job is logged on interactively at the computer, adTempus will run the job in that logon session. The user will be able to see the user interface for any programs run by the job. If the user has more than one interactive logon session, the program will run in the logon session that was created first. If the user is not logged on, the job will run in a Hidden session.

★ **Interactive required.** This option is the same as **Interactive if available**, except that the job will fail if no interactive session is available.

A program that is properly designed for "batch" execution should not rely on direct interaction with users, and therefore should not need to display a user interface. For such programs, use the "Hidden" session whenever possible. This provides more security, and eliminates the annoyance of program windows popping up while a user is logged in to the computer.

Regardless of what interaction level is selected, if you run the job manually while logged in to the computer, you can choose to have the job executed in your current logon session. This is useful if you need to be able to see the user interfaces of the job's programs during job debugging.

If the adTempus administrator has disabled interactive job execution, only the "Hidden" option will be available.

Hold Status

The **Hold Status** determines whether and how the job can be executed.

- **Enabled (not held).** The job can be run by Triggers, Responses, and on-demand. Checking or unchecking this box unchecks or checks the other three options. If this option is checked, the job is fully enabled for all execution types.
- **Hold triggers.** All Triggers defined for the job are ignored. Note: This option holds all triggers for the job, not just Schedule Triggers.
- **Prevent chained execution by other jobs.** Any Job Control Actions that target the job will be prevented from executing it. adTempus will log an error message in the job log for the calling job, indicating that the target job cannot be run
- **Prevent manual execution.** Users will be prevented from executing the job on-demand using the Run command or the adtexec utility.

The Hold Status replaces the Enabled option (and corresponding Hold/Release commands) used in prior versions of adTempus. The **Prevent chained execution** option replaces the **Allow job to be triggered by other jobs** option that previously appeared on the Triggers page.

History Retention

Specify how long adTempus should retain the history for this job. The default setting is specified in the Server Options window.

Tags

Enter tags to categorize the job if desired. Tags can be used to filter jobs displayed in some views in the Console (including the Job Monitor view).

Related Concepts

★ Job Properties 133

Documentation Page

Location: Job Properties window

Description/Notes

Enter any extended descriptive information or notes for this job. There is no limit on the length of the text.

Related Concepts

Job Properties 133

Variables Page

Location: Job Properties window

The **Variables** page lists all Job Variables for the job. A job inherits variables from the group and queue that it belongs to, and from the server-level variables. You can add, modify, or delete new variables, or modify inherited variables. Inherited variables cannot be deleted.

The variables defined here are inherited by the job's steps, triggers, conditions, etc.

Related Concepts

Job Properties 133

Job Recovery Page

Location: Job Properties window

Options on the **Recovery** page determine what adTempus should do if a job is interrupted or missed.

Run on startup if last run missed

If the last scheduled execution time for the job (based on schedule triggers defined for the job) was missed because the adTempus service was not running, adTempus will run the job when the service starts.

You can also configure adTempus to send a notification message in this case, by using a Response with event "Scheduled execution missed."

Restart

The Restart options allow adTempus to restart a job that was interrupted.

Run on startup if adTempus was shut down while the job was running

If the adTempus service was stopped while the job was running, adTempus will restart the job when the service is started.

Stopping the adTempus service does not terminate running programs. Your program may therefore still be running when adTempus goes to restart the job. When you use this option you should use process conditions, the Program Execution task's **Skip if already running** option, or some other safeguard to make sure that your scheduled tasks do not get re-executed when they should not.

Run on startup if adTempus terminated unexpectedly while the job was running

If the adTempus stopped unexpectedly while the job was running, adTempus will restart the job when the service is started. An "unexpected termination" is any abnormal termination of the adTempus service. This could be due to a failure of the service itself or to a system failure.

adTempus is not able to determine the reason for the abnormal termination. If the termination was due to a cause other than an unexpected system shutdown, your program may still be running when adTempus goes to restart the job. When you use this option you should use process conditions, the Process Execution task's **Skip if already running** option, or some other safeguard to make sure that your scheduled tasks do not get re-executed when they should not.

Restart From

This option determines the point from which the job will be restarted:

- The beginning (first step) of the job
- The beginning of the step that was executing when the failure occurred
- The beginning of the step following the step that was executing when the failure occurred

Pass most recent checkpoint to job

Regardless of what step execution begins with, adTempus can pass the task the most recent checkpoint that was set before the failure occurred. Your program or script can use this to determine where it should resume execution.

Related Concepts

Job Properties 133

Triggers Page

Location: Job Properties window

Options on the **Triggers** page determine when the job will be run.

The **Allow job to be triggered by other jobs** option that previously appeared on this page has been replaced by the Hold Status setting on the General page.

Triggers

The **Triggers** list lists the triggers that have been defined for the job. You can add, edit, or delete triggers. See the Triggers topic for information on the available trigger types.

Multiple Instances

The **Multiple Instances** options determine what adTempus should do if an instance of the job is already running when a new instance is triggered:

- **Execute (start a new instance)**. A new instance will be started. Use this option when it is acceptable for overlapping instances of a job to be running.
- **Do not execute**. The job will be skipped.

  When a job is skipped due to this setting, no instance is recorded in the job's execution history, but the job's Last Status is set to "Skipped."
- **Execute only if the previous instance(s) complete within the specified interval**. adTempus will wait for the delay you specify. If the previous instance(s) complete within that time, a new instance will be started. Otherwise the execution will be skipped (as in the previous option).
- **Wait for the specified interval, then execute regardless of whether previous instances have completed**. adTempus will wait for the delay you specify. At the end of that time, a new instance will be started, regardless of whether previous instances have finished.
- **Terminate prior instances, then run**. adTempus will terminate (abort) any other instances of the job that are running, then start the new instance.

When you start a job manually using the Execute command or from another job using a Job Control action, you have the option to override these settings and force a new instance to be started.

If the job uses the **Terminate prior instances** setting and you use the **Force a new instance of the job if necessary** option when starting the job, adTempus will start the new instance *without* terminating prior instances.

Update the Cycle ID for this job's scope when the job runs

When this option is checked, adTempus will update the Cycle ID for the job's group (or the nearest parent group that is configured to use Cycle IDs) each time the job is run.

When you are using Cycle IDs, you must configure one job with this option checked to run at the beginning of the cycle. This is what tells adTempus that a new cycle has begun. It updates the Cycle ID for the job group, and that Cycle ID is then assigned to this job and all others that run within the cycle scope.

Related Concepts

Job Properties 133
Triggers 212

Conditions Page

Location: Job Properties window

The **Conditions** page defines conditions that must be satisfied before the job is run.

Do not confuse Conditions with Triggers. Conditions are evaluated after the job has been triggered; they do not *cause*the job to run. See the Comparison of Triggers and Conditions topic for more information.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- **Execute only if all conditions are met.** The job is only executed if all of the listed conditions are met.
- **Execute if any condition is met.**The job is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- **Fail the job.** The job's steps are not executed; the instance's status is set to Failed.
- **Execute anyway.** The job is executed anyway.
- **Skip the job (do not report as a failure).** The job is not run, but it is not treated as a failure. An instance is recorded in the job's history, with a status of "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the job. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Related Concepts

Job Properties 133

Steps Page

Location: Job Properties window

The **Steps** page lists the steps to be executed for the job.

Step Execution Sequence

The **Step Execution Sequence** determines how steps should be executed in jobs with more than one step:

- **Execute steps in sequence; job stops if a step fails.** adTempus executes each job in sequence as long as all steps succeed. If a step fails, the job stops, and the job's status is "Failed."
- **Execute steps in sequence; job continues even if a step fails**. adTempus executes each job in sequence regardless of the outcome of individual steps. After all steps have been executed, the job's status is based on the status of the last step executed.
- **Execute the first step, but do not execute additional steps except as directed by Responses.** Use this option if you want to control the sequence of steps. When this option is selected, adTempus only runs the first step automatically. You must then use Job Control actions to direct the execution flow. The job's status is based on the status of the last step executed.

Regardless of which option you specify here, Job Control actions executed by Responses within the job can alter the flow of the job. For example, a Job Control Action may cause the job to end early, or may jump to a step out of the specified sequence.

Steps

The **Steps** list lists the steps that have been defined for the job. You can add, edit, delete, or reorder steps. Each step executes a single task; see the Tasks topic for information on the available task types.

Related Concepts

Job Properties 133

Job Resources Page

Location: Job Properties window

The **Resources** page defines resources that the job needs.

Resource Failure

Select the action that adTempus should take if one or more resources cannot be loaded.

- **Fail the job.** The Job will terminate immediately with a Failed status.
- **Run the job anyway.**adTempus will write a warning message to the job log but will execute the job anyway. Your tasks may fail to execute correctly if they depend on network resources that are not available. This option should be used only if your tasks can operate properly without the requested resources.

⭐Resources List

The **Resources** list lists the resources that have been specified for the job. You can add, edit, or delete resources (such as network drive mappings). See the Resources topic for information on the available resource types.

Related Concepts

Resources 90
Job Properties 133

Related Topics

Network Resource 416

Job Responses Page

Location: Job Properties window

The **Responses** page defines responses that should be executed for the job. These Responses are executed in addition to any Responses defined for individual steps .

A job may also inherit Responses from its Job Group and Job Queue, though those Responses are not listed here.

The **Responses** list lists the responses that have been specified for the job. You can add, edit, delete, or reorder responses. See the Responses topic for more information.

The following events are defined for jobs:

Event	Description
Beginning of Job	Occurs at the beginning of the job, before adTempus has evaluated any conditions or attempted to execute any steps.
End of job	Occurs at the end of the job, regardless of the job's outcome.
Conditions failed	Occurs if one or more conditions is not met (occurs only once regardless of how many conditions failed).
Job aborted	Occurs when the job is aborted (manually or as a result of a Job Control action).
Job failed	Occurs when the job is ending with a Failed status, for whatever reason.
Job restarted	Occurs when the job is being restarted by a Job Control action.
Job skipped because another instance is running	Occurs when another instance of the same job is already running and the Multiple Instances

★ Event	Description
	rule for the job specifies that the job should be skipped in this situation.
Job succeeded	Occurs when the job finishes with a successful result.
Restart limit exceeded	Occurs when the job is being restarted by a Job Control action but the restart limit specified on that action has been exceeded.
Execution time reached a threshold	Occurs if the job has been running for longer than the specified number of minutes.
Scheduled execution missed	Occurs at adTempus startup if one ore more scheduled executions were missed because adTempus was not running at the scheduled execution time. When this event is selected, the only available action is the Notification Action.
A trigger for the job failed	Occurs if there is a problem with one of the job's triggers. When this event is selected, the only available action is the Notification Action.
Job has been queued for longer than the specified time	Occurs if the job has been in the Queued for Execution or Waiting for Agent state for longer than the specified time. When this event is selected, the only available action is the Notification Action.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

Related Concepts

Job Properties 133
Responses, Events, and Actions 279

Related Topics

Actions 281
Events 280

Reference

Response Properties 280

Job Security Page

Location: Job Properties window

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to jobs:

Permission	Description
Full Control	Permission to perform all actions on the job.
List/Reference	Permission to link to the job in Job Control Actions, Job Conditions, etc. Does not automatically grant permission to view the properties of the job.
View	Permission to view the properties of the job.
Modify	Permission to modify the properties of the job. Includes the ability to hold and release the job.
Delete	Permission to delete the job.
Execute/Terminate	Permission to execute and abort the job through commands available in the console. Execute permission is also required for the user to link to the job using a Job Control Action.
Hold/Release	Permission to hold and release the job, but not to modify any of its other properties.
Administer security	Permission to change the security settings for the job.

To create a new job, a user must have permissions for both the group and queue that the job is being created in:

- For the group, the "Create jobs and groups in this group" permission.
- For the queue, the "Associate (assign jobs to this queue)" permission.

Related Concepts

Security 333

Related Topics

Security Login 339
Security Inheritance 335
Security Group 342
Security Editor 338
Initial Permission Setup 345
Administrator Provisioning 347
Permission Management Guidelines 343

Steps Overview

Each step of a job executes a single task to execute a program or take some other action for your job. Steps may additionally use Responses to respond to events such as program failure or to link to other jobs.

Steps for a job are managed through the Steps page of the job's properties.

The settings available for a step depend on the task that the step is configured to execute. For details, see the Tasks Overview or the help topic for the selected task.

Related Concepts

adTempus Objects	133
Key Concepts	81

Related Topics

adTempus Objects	133
Conditions	258
Triggers	212
Tasks	147

Reference

Job Properties	133

Checkpoints

adTempus provides the capability of restarting jobs that were interrupted due to a system shutdown or failure. In such cases, though, you may not want to start executing your task from the beginning in the event of a restart.

For example, if you have a batch file that executes several different commands, you may want to skip the commands that completed before the failure.

Checkpoints provide a means of doing this. As your program, batch file, or script executes, it can periodically pass a checkpoint value to adTempus (this can be any string that will be meaningful to your program). If adTempus subsequently restarts the job after a failure, it will pass back the latest checkpoint it received; your task can use that to resume execution at the correct place.

The method for getting and setting the checkpoint depends on the kind of task you are executing:

- Batch files should use the adtChkpt utility program.
- Scripts (as long as they are run by an adTempus Script Execution task) can use the Checkpoint variable.

- Applications can retrieve the checkpoint by using Windows API functions to retrieve the "ADTJobCheckpoint" environment variable. To set the checkpoint the application must use the programming interface to the adTempus engine.

Viewing Checkpoints

The History Panel and Job Monitor View both have a column that shows the most recent checkpoint for an active job. You can also view the most recent checkpoint of an active or completed instance in the Job Instance Properties window.

Example: Using Checkpoints in a Batch File

When you run a batch file using the internal batch file option in a Program Execution Task, adTempus can automatically insert checkpoints for you. When you use this option, you can leave out the calls to "adtchkpt" shown in the example below: adTempus will insert a call right after each label when it runs the batch file.

The following template shows how to use checkpoints effectively in a batch file.

```
@echo off
if not "%ADTJobCheckpoint%" == "" goto
%ADTJobCheckpoint%
:runimport
"%ADTServerPath%\adtchkpt" /s "runimport"
"c:\my programs\importdata.exe" 1234
rem Be sure to change both the label and the adtchkpt
parameter for each
rem checkpoint.
:createreports
"%ADTServerPath%\adtchkpt" /s "createreports"
"c:\my programs\createreports.exe" 1234
```

When adTempus runs a program or batch file with a resume checkpoint set, that checkpoint value is stored in the ADTJobCheckpoint variable. The second line of the batch file checks for that variable:

```
if not "%ADTJobCheckpoint%" == "" goto %ADTJobCheckpoint%
```

If the variable is set, the batch file jumps to the label that has the same name. Otherwise, the batch file continues execution from the beginning.

For each major command or section of your batch file, you define a label (to be used as the goto target if the step is rerun from that checkpoint) and pass the same value to adTempus as the latest checkpoint.

This line defines the label:

```
:runimport
```

The next line tells adTempus what the current checkpoint is:

```
"%ADTServerPath%\adtchkpt" /s runimport
```

The ADTServerPath environment variable is set by adTempus and points to the adTempus installation directory, where the adtchkpt program is located. Running that program tells adTempus what the current checkpoint is for the step.

Note that the label you use in the batch file and the checkpoint value you pass to adTempus must match.

Related Concepts

Program Execution Task 188

Reference

Program Execution Task Properties 189

Tasks

Tasks

Tasks represent the kinds of activities that adTempus is able to automate. Each step of a job executes a single task.

To perform actions in response to state changes during the execution of a job (for example, to send notification if a task fails, or link jobs together), use Responses instead of Tasks.

adTempus includes the following tasks:

Basic Tasks

- The Program Execution Task executes an external program, batch file, document, etc.
- The Script Execution Task executes an internal or external script.
- The Service Control Task starts, stops, or monitors a service.
- The Computer Shutdown Task shuts down or restarts the computer.
- The Process Termination Task terminates a running program.
- The Job Variable Update Task sets the value of a Job Variable.

Database Tasks

- The Database Operation Task executes commands against a relational database.

E-Mail and Notification Tasks

- The E-Mail Processing Task retrieves and processes e-mail messages from a mail server.
- The Notification Task sends notification messages using e-mail, SMS, instant messaging, etc.

File Tasks

File tasks perform operations on local or remote files.

- The File Transfer Task copies, moves, and deletes files.
- The File Compression Task compresses files.
- The File Uncompression Task uncompresses files

Network Tasks

- The Web Request Task requests a page or file from a Web server.

Additional Tasks

adTempus allows application developers to create their own task types, so additional task types may be available to you depending on the other software installed on your computer.

Related Concepts

Key Concepts 81

Related Topics

adTempus Objects 133
Conditions 258
Triggers 212
Steps Overview 145

Computer Shutdown Task

Computer Shutdown Task

The Computer Shutdown task allows you to shut down or restart the computer on which adTempus is running.

When shutdown is initiated, Windows displays a warning message on the computer. This message stays on top of all other windows and remains visible until the computer shuts down.

Once shutdown is initiated it can be aborted by running the following Windows command:

```
shutdown /a
```

Like all other adTempus tasks, the Restart Computer Task is executed in the security context of the user account specified for the job. Therefore that user must have the authority to shutdown/restart the computer, or the task will fail. To avoid running the job under an Administrator account, you can check the Use system context for certain operations option for the Credential Profile that is used for the job.

Related Concepts

Tasks 147
Steps Overview 145

Reference

Computer Shutdown Task Properties 149

Computer Shutdown Task Properties

The **Computer Shutdown Task Properties** window contains the settings for a job step that executes a Computer Shutdown Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- ★ Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not	Occurs if the job or a step within the job has been waiting for conditions

Event	Description
met within the specified time	for longer than the specified time.

The following special events are defined for the Computer Shutdown task:

Event	Description
Shutdown could not be initiated	Occurs if shutdown could not be initiated (for example, if the user account used for the job does not have the necessary permission).
Shutdown initiated	Occurs once shutdown has been successfully initiated.

Restart Options

Target Computer

Specify whether to restart the adTempus server or a remote computer. To target a remote computer, enter the name or IP address of the computer.

The user account under which the job is running must have the necessary permissions to shut down/restart the target computer.

After displaying warning message, wait __ seconds before shutting down

Specify the amount of time, in seconds, adTempus should wait between displaying the warning dialog and beginning the shutdown process. If the interval is set to 0, the shutdown begins immediately.

Restart the computer

Check this option if you want the computer to restart after it shuts down. If this option is not checked, the computer remains shut down.

Force all applications to close

Check this option to force all applications to close. If this option is not checked, the shutdown process may be held up by applications that do not close

For example, an application may display a confirmation dialog box before it closes. If the application is not forced to close, shutdown will not continue until the dialog box is acknowledged by a user.

Display the following additional information in the warning message

When shutdown is initiated, Windows will show a dialog box to all logged-in users. You may optionally enter a brief message to be displayed in the dialog box.

Related Concepts

Computer Shutdown Task 148

Database Operation Task

Database Operation Task

The **Database Operation** task allows you to perform any of the following operations against a relational database:

- Execute a job on the database server
- Execute database update scripts
- Select a single row/column value into a Job Variable
- Select datasets (recordsets) that can be saved to a file or passed to a script for processing

adTempus supports the following database servers:

- SQL Server
- Oracle
- MySQL
- Other databases with an OleDb driver installed

Reviewing Task Output

If your database operation fails, error information will be written to the Job Log. Regardless of whether the operation succeeds or fails, more detailed information (such as commands executed and number of rows updated) can be found in the Job Detail Log.

adTempus also records each command executed and the results returned in a temporary file that is available for the duration of the job. This file can be used by scripts or other processes within the job to examine the results of the task. After the step executes, the name of the file can be found in the "DatabaseTask.DatabaseResultFile" job variable. This file is also saved as a Captured File in the job history, with name "Database Results.txt".

Related Concepts

Tasks 147
Steps Overview 145

Reference

Database Operation Task Properties 152

Database Operation Task Properties

The **Database Operation Task Properties** window contains the settings for a job step that executes as Database Operation Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard Response Events, this task supports the following special events:

Event	Description
Value selected from database matches specified value	Applies only if the operation is "Select a scalar value into a Job Variable." The Response will run if the value returned by the query (i.e., the value assigned to the target Job Variable) matches the specified value.
The database job was canceled	Occurs if the database job being monitored by adTempus was canceled on the database server.

Database Connection

Database Type

Select the type of database server you want to connect to. adTempus supports the following database servers:

- **SQL Server.** No additional configuration is required.
- **Oracle.** You must have the Oracle client components and the Oracle OleDb provider (OraOLEDB.Oracle) installed on the adTempus server. See adTempus Knowledge Base article K00000535 for more information.
- **MySQL.** No additional configuration is required.

- **Other.** Other database types are supported if there is an OleDb driver for the database installed on the adTempus server. You must enter the OleDb connection string below.

Specify custom connection string

Check this option to enter a custom OleDb connection string for connecting to the database.

If you include the tokens `{userid}` and `{password}` in your connection string, they will be replaced with the user ID and password from the Credential Profile specified in the Database Authentication section. This allows you to avoid having the password saved in clear text.

Database Connection

If you have not checked **Specify custom connection string**, enter the values needed to connect to the database.

For Oracle databases, you can choose to **Use TNS**, in which case you enter the service alias defined in the tnsnames configuration file (remember, this must be configured on the computer where the adTempus service is running, if that is different from where you are running the Console). Alternatively, adTempus can connect without using a TNS alias: enter the **Host Name** and **Service Name** directly.

Database Authentication

Select **Use integrated (automatic) authentication** if you want the database server to authenticate automatically based on the Windows identity used for the job.

Select **Specify database credentials** to enter explicit credentials for the database server. The Credential Profile selector will filter the available Credential Profiles to only those defined for the database server you have selected.

Test Connection

Click **Test Connection** to verify that adTempus can connect to the database server using the settings you have provided.

Database Operation

Specify the database operation to perform.

By default adTempus uses a timeout value of 20 minutes for database selection and update operations (all operations other than job execution). That is, if your SQL execution or selection takes more than 20 minutes, the database will abort it and the task will fail.

If you need to allow operations to run longer than this, change the timeout value in the Server Options window.

Execute SQL

adTempus will execute the SQL commands you provide. To execute multiple commands, you should separate commands with a line containing only "/" or "go".

Select a scalar value into a Job Variable

adTempus will execute the SQL query you provide, which should be a scalar selection query (a query that returns a single value, such as "select count(*) from MyImportTable"). The result of that query will be stored in the Job Variable that you specify, making it available for other operations within the job.

If you specify more than one command in the SQL editor, only the result from the last statement executed will be stored in the variable.

Select data into a DataSet

adTempus will execute the SQL query you provide and return the selected data as a .NET DataSet. The resulting DataSet can be sent to:

- **A script.** Select or create a .NET script to execute. In the script, the returned data will be available in Parameters.DataSet, which will be a System.Data.DataSet object.
- **A file.** The DataSet will be written to the file in the DataSet XML format.

If you specify more than one command in the SQL editor, only the result from the last statement executed will be stored in the DataSet.

Execute database job

adTempus will execute a job defined on the database server (e.g., a job defined in SQL Server Agent on SQL Server). This operation is only available for SQL Server and Oracle databases.

Specify the name of the job to run. This value is not validated when you enter it; it must be the name of a job on the target database server.

If you check **Wait for job to complete**, the adTempus job step will continue to run as long as the database job is running, and the status of the step will reflect the outcome of the database job (succeeded or failed). If the target job is already running, adTempus will not start a new copy of it, but will monitor the already-running job.

If **Wait for job to complete** is not checked, adTempus will submit the job but will not wait for it to run. The status of the adTempus job step will be "Succeeded" if the database job was successfully started, or "Failed" if the job could not be submitted. If the target job is already running, adTempus will log an informational message in the adTempus Job Log and the step will be reported as "Succeeded."

Related Concepts

Database Operation Task........152

E-Mail Processing Task

E-Mail Processing Task

The **E-Mail Processing Task** allows you to retrieve e-mail messages from POP3 and IMAP servers and save the messages and/or message attachments to disk. Filtering rules allow you to select messages based on properties such as message contents, message headers, and attachment names.

The E-Mail Trigger provides the same e-mail processing functionality and can be used to poll the mail server and trigger the job when messages arrive. Use the E-Mail Processing Task when you want to process mail after triggering the job using a different kind of trigger.

For example, if you want to run a job each time an e-mail message arrives, you would use the E-Mail Trigger. If instead you wanted to run a job at 6:00PM to process any messages that have arrived during the day, you would use a Schedule Trigger to run the job, and an E-Mail Processing Task to download the messages.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

E-Mail Trigger 222

Reference

E-Mail Processing Task Properties 157

E-Mail Processing Task Properties

The **E-Mail Processing Task Properties** window contains the settings for a job step that executes an E-Mail Processing Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.

Event	Description
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard response events, this task defines the following events:

Event	Description
No messages selected	Occurs if no messages matched the selection criteria.
One or more messages selected	Occurs if at least one message matches the selection crtieria.
Selected message count matches a value	Occurs if the number of selected messages meets the specified criterion.

Message Source

Mail connection to use

Select an existing Incoming E-Mail Provider or create a new one. The e-mail provider represents a connection to a POP3 or IMAP mail server using a single login account.

IMAP Folder

For an IMAP connection you may optionally enter the name of the folder to select messages from. Leave blank to select from the default Inbox folder for the account.

Delete selected messages

If this option is checked, adTempus will delete from the mail server each message that is selected and processed by the e-mail processing operation. Messages that do not meet the selection criteria are not deleted.

If this option is not checked, adTempus will leave the messages on the server. However, adTempus remembers which messages it has already seen and will not reprocess the messages again in the future unless you clear the message list as described in the next section.

Some mail servers (such as Gmail) may not delete messages even if adTempus tells them to.

Clear list of previously-processed messages

Click this button to clear the list of messages that adTempus has already examined in this mailbox. This will cause adTempus to examine all messages in the mailbox the next time it connects to the server. Use this option if, for example, you have changed the message filtering rules and want adTempus to re-examine messages to select messages that were not processed in the past because they did not meet the selection rules.

Message Selection

This page allows you to define criteria controlling which messages adTempus will select for processing. If no criteria are defined, adTempus will select all new messages in the mailbox.

adTempus remembers which messages it has already seen, and will never re-read messages it has already seen, unless you clear its list using the **Clear list of previously-processed messages** button.

Selection Rules

If no rules are defined, adTempus will select all messages in the mailbox. To filter messages based on criteria based on their subject, content, sender, attachments, etc., add one or more selection filters.

If you add more than one rule, specify whether adTempus should select messages that match **Any** rule or only messages that match **All** rules.

Selection Script

For more advanced filtering, you can use a .NET script to filter the message. For each message that meets the basic selection criteria (if any), adTempus will run your script. Your script has access to the Parameters.Message object, which exposes the message and its attachments through a MailMessage object.

Your script must return True if adTempus should select the message or False if it should not.

Message Saving

The options on the Message Saving page allow you to save selected messages and/or their attachments to disk.

Save format

Select the format to save the message in:

- **RFC 822.** The message is saved in standard RFC 822 format with file name `message.eml`. Any attachments are included in the file as embedded MIME sections.
- **RFC 822 with attachments.** The message is saved in standard RFC 822 format with file name `message.eml`, and attachments are also saved in separate files.
- **HTML with attachments.** The body of the message is saved to an HTML file named `message.htm`. Attachments are saved in separate files.

- **Text with attachments.** The body of the message is saved to a text file named `message.txt`. Attachments are saved in separate files.
- **Attachments only.** The body of the message is not saved, but all attachments are saved in separate files.

Save messages to folder

Specify the base folder where messages should be saved. adTempus will create a uniquely-named folder under this folder for each message. The folder will contain a file for the message body (named as specified above based on the **Save format**) and/or the attachments for the message.

adTempus will store a list of the folders that it creates in the **EMailProcessing.MessageLocation** Job Variable.

For example, you set the save location to `c:\messagedownload`. adTempus processes 2 messages and saves them in format **HTML with attachments**. The variable will be set to:

```
c:\messagedownload\{C7C9286D-607F-4192-9960-
EDED8624A4BE};c:\messagedownload\{837FC8D4-314D-41E3-B95E-412CB84694E5}
```

The messages can be found in files `c:\messagedownload\{C7C9286D-607F-4192-9960-EDED8624A4BE}\message.htm` and `c:\messagedownload\{837FC8D4-314D-41E3-B95E-412CB84694E5}.htm`.

The attachments for each message can be found in the corresponding folders.

Job Variables

This task creates the following Job Variables, which are available for the remainder of the job:

Name	Description
EmailProcessing.MessageLocation	A semicolon-delimited list of the files or directories created if messages or attachments were saved. See the Message Saving section for more information.
EmailProcessing.SelectedMessageCount	The number of messages selected by the task (the number of messages that meet the selection criteria).

Related Concepts

E-Mail Processing Task 157

File Operation Tasks

adTempus includes the following tasks that can operate on local or remote files:

- The File Transfer Task copies, moves, and deletes files.
- The File Compression Task compresses files.
- The File Uncompression Task uncompresses files

Related Topics

File Trigger 228
File Condition 260

File Transfer Task

File Transfer Task

The File Transfer task allows you to copy, move, or delete files located on the Windows file system or on FTP or SFTP file servers.

After a File Transfer task executes, detailed information about the file operation can be found in the Job Detail Log.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

Web Request Task 208

Reference

File Transfer Task Properties 162

File Transfer Task Properties

The **File Transfer Task Properties** window contains the settings for a job step that executes a File Transfer Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.

★ Event	Description
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard Response Events, this task defines the following events:

Event	Description
Affected file count matches a value	The number of files transferred/deleted matches the value specified.
No files affected	No files were transferred/deleted.
One or more files affected	At least one file was transferred/deleted.

Source

The source page contains options for selecting the files to copy, move, or delete.

Action

Select the action you want to perform: Copy, Move, or Delete files.

Location

Select the location for the source files. If the files are located on the adTempus server or another computer reachable using a standard UNC share, select **Local or Network File**.

If the files are located on a remote file server (such as FTP or SFTP), select the file server or create a new one. This defines the settings needed for adTempus to connect to the file server.

Source Root

The Source Root option determines how you will specify the files to include.

Specify all files/paths relative to a single directory

If you choose **Specify all files/paths relative to a single directory**, you will enter each file name relative to the root folder you define.

For example, suppose all the files you wish to transfer are located below the "c:\sourcefiles" directory. You would specify "c:\sourcefiles" as the source root here. In the **Include Files** list you would specify the files relative to that location, e.g.:

file_a.txt
file_b.txt
subdir\file_c.txt

The root directory name may include job variables.

Include all subdirectories

If you check **Include all subdirectories**, adTempus will search file files that match the patterns you specify in the **Include Files** list in the source root directory and all of its subdirectories. If you do not check this option, subdirectories will not be searched unless you include them explicitly in the **Include Files** list.

Specify absolute paths

If you choose **Specify absolute paths**, you will enter the full path for each file. From the example above, you would specify the **Include Files** as:

c:\sourcefiles\file_a.txt
c:\sourcefiles\file_b.txt
c:\sourcefiles\subdir\file_c.txt

ℹ If you want to keep the source directory structure during the transfer, you must use the **Specify all files/paths relative to a single directory** option.

Include Files

Specify the names or patterns of the files to include.

If you selected the **Specify all files/paths relative to a single directory** option, each entry must be a file name only or a path relative to the source root; it cannot start with "\", "/", or a drive letter.

If you selected the **Specify absolute paths** option, each entry must include a full path to the file.

You can use wildcards to match file and directory names, and the names may include job variables.

Exclude Files

Specify names or paths to exclude from the selection. You may specify full paths, partial paths, file names only, or wildcards. Job variables can be used.

Examples:

sometext
  Excludes any file with "sometext" anywhere in its name or path
\sometext
  Excludes any file with "sometext" as one of the nodes in its path
**.bak*

Excludes any file with file extension ".bak"

sometext

Excludes any file with "sometext" anywhere in its name or path

*c:\**\backup*.**

Recursively excludes all files in any directory named "backup" that appears anywhere below "c:\"

Destination

The Destination page contains options specifying the destination for copy and move operations.

Location

Select the location for the target files. If the target location is on the adTempus server or another computer reachable using a standard UNC share, select **Local or Network File**.

If the target location is on a remote file server (such as FTP or SFTP), select the file server or create a new one. This defines the settings needed for adTempus to connect to the file server.

Transfers from a remote location to a remote location are not supported. Such an operation must be performed in two steps: first download the files to a local temporary directory, then upload them to the target server.

Destination Directory

Specify the base directory for the files to be transferred to.

Preserve source directory structure

Check this option to preserve the relative directory structure of the source files. This option is available only if you checked the **Specify all files/paths relative to a single directory** option on the **Source** page.

If **Preserve source directory structure** is not checked, all files will be written to the root **Destination Directory**, regardless of what their original path was.

File Overwrite

Specify what adTempus should do if the target file already exists:

- **Skip the file.** The file will not be transferred.
- **Overwrite the existing file.** The file will be replaced.
- **Overwrite the existing file if it is older.** The file will be replaced if it is older than the file being transferred.

  Note: This option may not be reliable for files being transferred to or from an FTP server, since adTempus cannot always accurately determine the time zone for a file's timestamp when it is on an FTP server.

- **Use a unique name for the new copy.** The existing file will be preserved, and the source file will be copied under a new name. The new name will be created by inserting a timestamp in the original name. For example, `afile.txt` will be copied as `afile.20130501184244.txt`.

Make backups of existing files before replacing

Check this option to create a backup of any file that will be replaced during the operation.

The backup name will be created by inserting a timestamp in the original name. For example, `c:\target\afile.txt` will be backed up to `"c:\target\afile.backup_20130501184244.txt`.

Job Variables

This task sets the following Job Variables, which can be used by subsequent operations within the job:

Variable	Description
FileOperationTask.ProcessedFileCount	The number of files transferred/deleted.

Related Concepts

File Transfer Task 162

File Compression Task

File Compression Task

The File Compression task allows you to compress files located on the Windows file system or on FTP or SFTP file servers. You can use the File Uncompression Task to extract files from a compressed archive.

Currently the task supports the ZIP compression standard.

After a File Compression task executes, detailed information about the file operation can be found in the Job Detail Log.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

File Uncompression Task 172

Reference

File Compression Task Properties 168

File Compression Task Properties

The **File Compression Task Properties** window contains the settings for a job step that executes a File Compression Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

★Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard Response Events, this task defines the following special Events:

Event	Description
Affected file count matches a value	The number of files compressed matches the value specified.
No files affected	No files were compressed.
One or more files affected	At least one file was compressed.

Source

The **Source** page contains options for selecting the files to compress.

Location

Select the location for the source files. If the files are located on the adTempus server or another computer reachable using a standard UNC share, select **Local or Network File**.

If the files are located on a remote file server (such as FTP or SFTP), select the file server or create a new one. This defines the settings needed for adTempus to connect to the file server.

Source Root

The Source Root option determines how you will specify the files to include.

Specify all files/paths relative to a single directory

If you choose **Specify all files/paths relative to a single directory**, you will enter each file name relative to the root folder you define.

For example, suppose all the files you wish to transfer are located below the "c:\sourcefiles" directory. You would specify "c:\sourcefiles" as the source root here. In the **Include Files** list you would specify the files relative to that location, e.g.:

```
file_a.txt
file_b.txt
subdir\file_c.txt
```

The root directory name may include job variables.

Include all subdirectories

If you check **Include all subdirectories**, adTempus will search file files that match the patterns you specify in the **Include Files** list in the source root directory and all of its subdirectories. If you do not check this option, subdirectories will not be searched unless you include them explicitly in the **Include Files** list.

Specify absolute paths

If you choose **Specify absolute paths**, you will enter the full path for each file. From the example above, you would specify the **Include Files** as:

```
c:\sourcefiles\file_a.txt
c:\sourcefiles\file_b.txt
c:\sourcefiles\subdir\file_c.txt
```

If you want to store path information in the compressed archive, you must use the **Specify all files/paths relative to a single directory** option.

Include Files

Specify the names or patterns of the files to include.

If you selected the **Specify all files/paths relative to a single directory** option, each entry must be a file name only or a path relative to the source root; it cannot start with "\", "/", or a drive letter.

If you selected the **Specify absolute paths** option, each entry must include a full path to the file.

You can use wildcards to match file and directory names, and the names may include job variables.

Exclude Files

Specify names or paths to exclude from the selection. You may specify full paths, partial paths, file names only, or wildcards. Job variables can be used.

Examples:

sometext
Excludes any file with "sometext" anywhere in its name or path
\sometext
Excludes any file with "sometext" as one of the nodes in its path
**.bak*
Excludes any file with file extension ".bak"
sometext
Excludes any file with "sometext" anywhere in its name or path
*c:\**\backup*.**
Recursively excludes all files in any directory named "backup" that appears anywhere below "c:\"

Delete files after they are added to the archive

Check this option to delete the source files after the archive has been created successfully.

Destination

The Destination page contains options specifying how and where the files will be compressed.

Compression Type

Select the compression standard to use.

Location

Select the location for the target files. If the target location is on the adTempus server or another computer reachable using a standard UNC share, select "Local or Network File."

If the target location is on a remote file server (such as FTP or SFTP), select the file server or create a new one. This defines the settings needed for adTempus to connect to the file server.

Compress To

Specify the name of the archive to create.

Store path information in archive

Check this option to save the relative path information for each file in the archive. This option is available only if you checked the **Specify all files/paths relative to a single directory** option on the Source page.

Existing Archive

Specify what adTempus should do if the target file already exists:

- **Replace (overwrite) it Skip the file.** The existing archive will be replaced with the new copy.

- **Add to/update it.** The existing archive will be updated with the source files. adTempus will add and update files in the archive as appropriate. It will not delete any files from the archive.

Make a backup of the existing file before replacing or updating it

Check this option to create a backup of the archive before replacing or updating it.

The backup name will be created by inserting a timestamp in the original name. For example, `c:\target\myfiles.zip` will be backed up to `c:\target\myfiles.backup_20130501184244.zip`.

Encryption

If you wish to encrypt the archive, specify the encryption type and password.

The available encryption types will depend on the compression standard being used.

For ZIP compression, the following encryption options are available:

- **Standard pkzip encryption.** This is the original encryption method used by pkzip and is considered to be weak encryption. This encryption method is supported by most ZIP utilities.
- **WinZIP AES 128-bit encryption.** Uses 128-bit AES encryption. This encryption method may not be supported by all ZIP utilities.
- **WinZIP AES 256-bit encryption.** Uses 128-bit AES encryption. This encryption method may not be supported by all ZIP utilities.

Job Variables

This task sets the following Job Variables, which can be used by subsequent operations within the job:

Variable	Description
FileOperationTask.ProcessedFileCount	The number of files compressed.

Related Concepts

File Compression Task........167

File Uncompression Task

File Uncompression Task

The File Uncompression task allows you to uncompress files from compressed archives such as ZIP files. You can use the File Compression Task to create compressed archives.

Currently the task supports the ZIP compression standard.

After a File Uncompression task executes, detailed information about the file operation can be found in the Job Detail Log.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

File Compression Task 167

Reference

File Uncompression Task Properties 173

File Uncompression Task Properties

The **File Uncompression Task Properties** window contains the settings for a job step that executes a File Uncompression Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard Response Events this task defines the following special Events:

Event	Description
Affected file count matches a value	The number of files extracted matches the value specified.
No files affected	No files were extracted.
One or more files affected	At least one file was extracted.

Source

The **Source** page contains options for selecting the archives to extract files from.

Location

Select the location for the source archives. If the files are located on the adTempus server or another computer reachable using a standard UNC share, select "Local or Network File."

If the files are located on a remote file server (such as FTP or SFTP), select the file server or create a new one. This defines the settings needed for adTempus to connect to the file server.

Files to uncompress

Enter the names of the archives to uncompress. You may uncompress multiple archives: enter one archive name per line. Each entry must be the full path and name of the archive. Wildcards and job variables can be used.

You are specifying here the name of the archive (e.g., the ZIP file). adTempus will extract all files from each specified archive.

Compression Type

Select the compression type used for the archives. Generally you can leave this set to **Auto Detect**, but you must select the Compression Type if you need to specify as password below for decryption.

Delete archives after files are extracted

Check this option to delete archives after the files have been extracted from them. Archives will only be deleted if all files are successfully extracted from them.

Encryption

If the archive is protected by encryption, select the encryption type and enter the password here.

This option is only available if you have selected a Compression Type.

Destination

The Destination page contains options specifying where the files will be uncompressed.

Location

Select the location for the target files. If the target location is on the adTempus server or another computer reachable using a standard UNC share, select **Local or Network File**.

If the target location is on a remote file server (such as FTP or SFTP), select the file server or create a new one. This defines the settings needed for adTempus to connect to the file server.

Uncompress To

Specify the base directory where the files will be extracted.

Create a subdirectory for each archive

If you specified more than one archive on the **Source** page, check this option to have adTempus create a subdirectory for each archive. If this option is not checked, all files from all archives will go to the same directory.

Use path names from archive

Check this option to have adTempus recreate the directory structure stored in the archive. If you do not check this option, or if the archive does not include path information, all files will be written to the same directory.

Check this option to save the relative path information for each file in the archive. This option is available only if you checked the **Specify all files/paths relative to a single directory** option on the Source page.

File Overwrite

Specify what adTempus should do if the target file already exists:

- **Skip the file.** The file will not be transferred.
- **Overwrite the existing file.** The file will be replaced.
- **Overwrite the existing file if it is older.** The file will be replaced if it is older than the file being transferred. Note: This option may not be reliable for files being transferred to or from an FTP server, since adTempus cannot always accurately determine the time zone for a file's timestamp when it is on an FTP server.
- **Use a unique name for the new copy.** The existing file will be preserved, and the source file will be copied under a new name. The new name will be created by inserting a timestamp in the original name. For example, `afile.txt` will be copied as `afile.20130501184244.txt`.

Make backups of existing files before replacing

Check this option to create a backup of any file that will be replaced during the operation.

The backup name will be created by inserting a timestamp in the original name. For example, `c:\target\afile.txt` will be backed up to `c:\target\afile.backup_20130501184244.txt`.

Job Variables

This task sets the following Job Variables, which can be used by subsequent operations within the job:

Variable	Description
FileOperationTask.ProcessedFileCount	The number of files uncompressed.

Related Concepts

File Uncompression Task 172

File Selection Wildcards

adTempus supports the following standard wildcards for file name matching:

?
Matches exactly one character

*
Matches 0 or more characters

When used in paths, the * wildcard behaves as follows:

*\
Matches any subdirectory

**\
Recursively matches any subdirectory, or no subdirectory at all.

The following examples demonstrate wildcard matching for directories:

c:\source\*\files*.txt
Match: c:\source\foldera\files\afile.txt
Match: c:\source\folderb\files\afile.txt
No Match: c:\source\foldera\subfolder\files\afile.txt
No Match: c:\source\files\afile.txt

c:\source\*\files*.txt*
Match: c:\source\files\afile.txt
Match: c:\source\foldera\files\afile.txt
Match: c:\source\foldera\subfolder\files\afile.txt
Match: c:\source\folderb\files\afile.txt
No Match: c:\source\foldera\subfolder\files\subdir\afile.txt
No Match: c:\source\files\afile.txt

c:\source\**.txt*
Match: c:\source\afile.txt
Match: c:\source\foldera\afile.txt
Match: c:\source\foldera\subfolder\afile.txt

Job Variable Update Task

Job Variable Update Task

The **Job Variable Update** task allows a job to create or update a Job Variable. The update may affect the current instance of the job only, or can be made permanent.

The same functionality is also available using the Job Variable Update Action, which can be executed from Responses.

Related Concepts

Tasks 147
Steps Overview 145
Job Variables 88

Related Topics

How To: Set and Retrieve Variables in Scripts 500
Job Variable Condition 269
Job Variable Update Action 289
Predefined Variables 485
Text Edit Window 441

Reference

Job Variable Update Task Properties 178
Job Variable Properties 412

Job Variable Update Task Properties

The **Job Variable Update Task Properties** window contains the settings for a job step that executes a Job Variable Update Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.

Event	Description
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

Variable Update

Variable to Update

Enter the name of the Job Variable to update. Click the **...** button to select from a list of already-defined variables. If the variable already exists, the existing value will be updated. Otherwise a new variable will be created.

Note: Do not include percent signs ("%") around the variable name.

Set value to

Select this option to set the variable to a specific value. Enter the value you want to set the variable to.

Check the **Expand variable tokens at runtime** option if you want to expand any variable tokens found in the value you have specified. For example, suppose you are updating a variable named "BaseVariable" and you specifiy that it should be set to `%Variable22%`. If **Expand variable tokens at runtime** is checked, BaseVariable will be set to whatever the value of Variable22 is at runtime. If the option is not checked, BaseVariable will be set to the literal value `%Variable22%` `Variable22` will not be treated as a variable name).

Increase/decrease value by

Select this option to increase or decrease a numeric variable by the amount you specify. If the target variable already exists, it must contain a numeric value type or the update will fail. If the target variable does not exist, it will be created and set to the increment value.

Scope

The **Scope** determines how broadly the variable update will apply.

- **Apply to executing instance only.** The new value of the variable will be seen only within the current instance of the job.

For the remaining settings the new value of the variable will be seen within the current instance of the job, and the definition will also be updated in the variable's containing object, so that future job executions will also be affected.

ℹ The user who is editing the settings must have permission to modify the target object or the settings cannot be saved. No permission check is made at job execution.

ℹ If auditing is turned on for the affected object, adTempus will generate a change log entry for the variable update. You can change this behavior.

- **Update definition for Step.** The variable is updated in the definition of the Step that is running the update.
- **Update definition for Job.** The variable is updated in the definition of the Job that is running the update.
- **Update definition for Job Group.** The variable is updated in the definition of the Job Group that the executing job belongs to. If the variable is already defined for a group in the job's ancestor hierarchy, the variable will be updated at that level. Otherwise, the variable will be created in the Group that is the job's immediate parent.
- **Update definition for Server.** The variable is updated at the server level.

Related Concepts

Job Variable Update Task 178

Notification Task

Notification Task

The **Notification Task** allows you to send notification messages, and optionally capture and attach files.

The Notification Action and File Capture Action provide similar functionality. Those actions can only be run as Responses for a job or a step, but the Notification Task, because it is a task, can be the primary function that a job or step performs.

Generally, you would use a Notification or File Capture Action when you want to send a notification message in response to something that occurs during job execution. You would use Notification Task when the notification message is the purpose of the job or step.

Unlike the Notification Action and File Capture Action, the Notification Task can send messages to e-mail recipients who are not formally defined in adTempus as Notification Recipients.

Related Concepts

Tasks 147

Steps Overview 145

Related Topics

Notification Action 291
Notification Recipients 93
Messaging Setup 417
Notification Recipient 303
Notification Group 309
How To: Send Notification Messages for Failed Jobs 493

Reference

Notification Task Properties 182

Notification Task Properties

The **Notification Task Properties** window contains the settings for a job step that executes aNotification Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

⭐If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

⭐Notification Message

Subject of message

Provide a subject for the message that will be sent.

The subject may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

Message to send

Specify the message to be sent.

The message may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

Notification Severity

Specify the severity (importance) of this message. Notification Recipients, Addresses, and Groups can all be configured to receive only messages that meet specified severity criteria.

E-Mail Sender

Optionally specify the display name or e-mail address to be used as the sender of e-mail notification messages. If you leave a box empty, adTempus will use the value specified in the SMTP server properties.

You can also override the default values by setting the **NotificationFromName** and/or **NotificationFromAddress**Job Variables for a job, group, queue, step, etc. You can do this either through the properties window for the job, group, etc., or by running a script within the job.

Recipients

Recipients Defined in adTempus

Add, edit, or remove the notification recipients who will receive the notification message.

Additional Recipients

You may define additional recipients who will receive notification by e-mail, without first defining them as Notification Recipients.

Attachments

The Attachments page allows you to specify files that will be attached to the notification message. In the Files list you can add, edit, or remove File Specifications that define the files to be attached.

Save these files in the job history

If this option is checked, the files you specify will also be captured and saved as part of the job's history. If the option is not checked, the files are sent, but are not saved with the job history.

Include captured console output

If this option is checked, any captured console output files from previous steps will be attached to the message.

Related Concepts

Notification Task	181

Process Termination Task

Process Termination Task

The Process Termination task allows you to have adTempus terminate a process that is executing outside of adTempus.

If the process is running as a service process, use the Service Control Task instead to stop the service cleanly.

Be sure you understand how adTempus terminates a process before using this task. Terminating a process is not always "clean," and this can lead to problems (such as data corruption) with your application.

Related Concepts

Tasks	147
Steps Overview	145

Related Topics

Process Condition	272
Process Trigger	240
Service Control Task	204

Reference

Process Termination Task Properties	197

Process Termination Task Properties

The Process Termination Task Properties window is shown when you edit a Job Step that executes a Process Termination Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

⭐All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard response events, this task defines the following events:

Event	Description
No processes found to terminate	Occurs if no matching processes were running.
One or more proccess(es) could not be terminated	Occurs if one or more instance of the process could not be terminated.
One or more process(es) terminated	Occurs if one or more instance of the specified process was terminated.

Process Termination

Process to terminate

Specify the executable file name of the process that adTempus should terminate. You must specify the executable name (e.g., "notepad.exe"). If there is more than one program on the computer with the same executable file name, you may optionally include the full path of the version of the process that you want to monitor.

Note that this must be an executable (.exe). The task cannot target batch files (.bat, .cmd), scripts (.vbs, .js), etc.

If more than one instance of the process is running, all instances will be terminated.

Also terminate any child processes of the target process

Check this option to also terminate any processes that the target process has launched.

Job Variables

The Process Termination Task sets the following Job Variables, which can be used by subsequent operations in the job:

Variable	Description
ProcessTermination.KilledProcessCount	The number of processes matching the target process name that were successfully terminated.
ProcessTermination.FailedToKillProcessCount	The number of processes matching the target process name that were not successfully terminated.
ProcessTermination.KilledChildProcessCount	The number of processes terminated that were child processes of target processes.
ProcessTermination.FailedToKillChildProcessCount	The number of processes not terminated that were child processes of target processes.
ProcessTermination.KilledChildProcesses	Comma-separated list of the names of the child processes that were terminated.
ProcessTermination.FailedToKillChildProcesses	Comma-separated list of the names of the child processes that could not be terminated.

Related Concepts

Process Termination Task 196

Program Execution Task

Program Execution Task

The **Program Execution Task** is likely the main task that you will use in adTempus. The Program Execution task allows you to run programs and batch files.

Though the Program Execution Task can be used to run scripts as well, the Script Execution Task is tailored to script execution and is often a better choice. It can be used for PowerShell scripts as well.

The program is executed under the user account specified for the job, and therefore will only have access to the files and other resources that the user has access to. This includes the target program itself: the user must have the necessary permissions to launch the program.

Note that scheduled programs do not have access to network drive letters unless those drive letters have been mapped either by adTempus (on the Resources page for the job) or by the target itself (e.g., using a "net use" command in a batch file). See the Network Access topic for more information.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

Calling Shared Batch Files 195
Checkpoints 145

Reference

Program Execution Task Properties 189

Program Execution Task Properties

The **Program Execution Task Properties** window is displayed when you edit a Job Step that runs a Program Execution Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.

⭐Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard response events, this task defines the following events:

Event	Description
Task completed with an exit code that meets these criteria	Occurs when the target program finishes and returns a result that matches the criteria you specify.
Task executed but returned failure result	Occurs when the target program finishes and, based on the **Success Criteria** on the Advanced page, is determined to have failed.
Task returned successful result	Occurs when the target program finishes and, based on the **Success Criteria** on the Advanced page, is determined to have succeeded.
Task started	Occurs once the target program has been successfully started. Does not occur until after adTempus has waited as directed in the **Startup Determination** options on the Advanced page.
Task will be terminated because it exceeded its allowed execution time	Occurs once the time limit specified on the Advanced page has elapsed, but before adTempus attempts to terminate the process. Respond to this event if you want to use your own script or other approach to end the process.
Task was terminated because it exceeded its allowed execution time	Occurs once the time limit specified on the Advanced page has elapsed, after adTempus has terminated the process.
Execution time reached a threshold	Occurs when the program runs for longer than the specified time.

Execution Target

Target Type

The Program Execution Task supports three kinds of target:

- **Single program or external batch file.** You specify a single target for adTempus to execute. This can be a program, batch file, document file, etc.
- **Batch file stored in adTempus.** When you select this option, adTempus displays an editor that allows you to create a standard Windows batch file that is stored in adTempus. When the step executes, the batch file is executed just like a normal batch file. Using this option instead of an external batch file makes it easier to track changes to the batch file and to make sure the batch file gets included if the job is deployed to another computer.
- **Shared batch file stored in adTempus.** When you select this option, you can select or create a shared batch file. This allows you to reuse batch files in different jobs and steps. Shared batch files can be managed from the Shared Scripts view.

Single Program Target Settings

These settings apply if you choose to execute a single target.

Target

Specify the program, batch file, script file, document file, etc., to be executed. You may specify here essentially anything that is valid when using the **Run** command from the Windows **Start** menu. You do not need to include quotation marks around the target: they will be added for you if necessary.

Only specify the program or document path and name here. Any parameters or options that need to be passed to the program must be entered in the command-line parameters box below.

For example:

`c:\program files\serverdatacopy\prod1.exe`	Runs the "prod1.exe" program.
`c:\data\mydocument.doc`	Launches Microsoft Word and loads `mydocument.doc`.
`c:\cmd\copydata.cmd`	Runs the `copydata.cmd` batch file.

Command-Line Parameters

Specify any command-line parameters that should be passed to the program.

There are two ways to insert dynamic data into the command line:

- Use a script (see next section) to build the command line
- Use Job Variables or Inline Functions within the Command-Line Parameters box.

Use a script to specify command-line parameters

Check this option to use a script to set the command-line parameters. For example, you can use a script to pass the names of all the files found in a specific directory to the target application.

See the Script Overview topic for information on working with scripts.

Skip this step if the process is already running

When this option is checked, it is like having a Process Condition on the target process. If the target process is already running (regardless of whether it is under the control of adTempus), the step will be skipped.

ℹ This option is only valid when you have specified an executable file (.exe) as the **Target**. The option will not work with other kinds of targets (such as .bat or .cmd files, or documents).

Batch File Target Settings

If you choose the batch file target option, you will see an editor in which you can edit your batch file. The batch file must use standard Windows batch file syntax.

For a new step, adTempus will fill the editor with a template batch file that demonstrates how to use checkpoints in your batch file. When you use checkpoints, the job status will show what the most recent checkpoint was during execution, and you can start or restart the batch file from any checkpoint.

When you check the **Automatically insert checkpoints** option, adTempus will automatically insert a checkpoint after each label (line beginning with ":") that it finds in the batch file. If this option is not checked, you must insert your own checkpoints as described in the checkpoints topic.

A batch file can call other shared batch files during execution. To insert a reference to another batch file, place the cursor at the point in the batch file where you would like to call the other batch file, then select the target file from the **Call shared batch** drop-down list. See the Calling Shared Batch Files topic for more information.

Shared Batch File Target Settings

If you choose the shared batch file target option, you will see a selector that allows you to select an existing shared batch file or create a new one. Shared batch files can be managed from the Shared Scripts view.

When you check the **Automatically insert checkpoints** option, adTempus will automatically insert a checkpoint after each label (line beginning with ":") that it finds in the batch file. If this option is not checked, you must insert your own checkpoints as described in the checkpoints topic.

Common Settings

These settings apply to single-target and batch file targets.

Startup Directory

Specify the startup ("working") directory to be associated with the program. For most programs, it is not necessary to specify this directory.

Capture screen output from console-mode program

This option can be used with console-mode programs only. When this option is selected the program is run on a hidden desktop (and so is not visible to any user who is logged in to the computer). The output that the program writes to the console is captured in a file, which is stored in the execution history for the job. To review the output, view the properties for the instance once the step completes. The console output will be found in the Captured Files list for the job, with the name `console output.txt`.

Execution Priority

Specify the priority that the operating system should assign to the process.

Advanced

Limit execution to __ seconds

When this option is checked, adTempus will terminate the process if it is still running after the specified number of seconds.

This option should generally be used only as a last resort to terminate an application that is not behaving properly. adTempus cannot always terminate the process cleanly, and this can lead to problems (such as data corruption) with your application. See How adTempus Terminates a Process for more information.

If the task being executed is a batch file, adTempus terminates the batch file and any program(s) started by the batch file. This is a change from adTempus 3 and earlier, which did not terminate processes created by the batch file.

Before the process is terminated adTempus executes any Responses that include the "Before process is killed" event.

Startup Determination

The **Startup Determination** option can be used to allow an application to finish initializing before job execution continues. This option is useful when a subsequent action or step needs to interact with the program. Three options are available:

- **As soon as it is launched**. adTempus does not wait at all. As soon as it has launched the program, it moves on.
- **Once the program is waiting for user input.** adTempus waits until the program has completed initialization and its Windows message loop is waiting for messages.
- After __ seconds. Specify a delay, in seconds.

Success Criteria

The Success Criteria determine whether the step is reported as Successful or Failed. Four options are available:

- **Decide based on the program's exit code.** adTempus will look at the exit code returned by the process and evaluate it based on the rule you specify. If the exit code meets the criteria, the step will be reported as Successful; otherwise it will be reported as Failed.

  Many—but not all—programs return an exit code to indicate their status. By convention, an exit code of 0 indicates success; an exit code greater than 0 indicates failure.

  Note that the meaning of a particular exit code is dependent on the program being run. adTempus does not know what if anything an exit code means—it can only detect and respond to the exit code.

  For more information see the Exit Codes topic.

- **Report the step as successful unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task was successful. You can change the status of the step using a Job Control Action.

- ★ **Report the step as Failed unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task failed. You can change the status of the step using a Job Control Action.

- **Use a script.** adTempus will ignore the exit code and will execute the script that you specify. The result of this script will determine whether the step succeeded.

  This feature can be used, for example, when you are running a program that does not produce a meaningful exit code, but produces a file if it succeeds. You could use a script to check for the existence of the file and set the step's status accordingly.

ℹ You can also test for any number of specific exit codes and take action based on them using Responses for the step. For example, you may want to report the step as failed if the exit code is greater than 0, but send e-mail notification to an administrator if the exit code is greater than 128.

Related Concepts

Program Execution Task 188

Related Topics

Calling Shared Batch Files 195
Checkpoints 145

Calling Shared Batch Files

This feature is new for adTempus version 4.

When you run a batch file stored in adTempus using the Program Execution Task, you can make other batch files stored in adTempus available for your batch file to call.

To do this, the batch file you want to call must first be created as a shared batch file (see below).

The batch file editor includes a **Call shared batch** drop-down list that includes all shared batch files. When you click this button, the editor will insert a batch call command at the cursor location in the editor to call the selected shared batch file. It also adds a link to tell adTempus to make that batch file available at runtime.

When the job executes, any shared batch files referenced by your batch file are written to a temporary folder whose path is stored in the ADTJobTemp environment variable.

Sharing a Batch File

New Batch File

If you are creating a new batch file and want it to be shared (available to other jobs), you can create the batch file in either of these ways:

- From the Shared Scripts view.
- From the Program Execution Task properties window. Select the shared batch file target type, then click the button to create a new batch file.

Sharing an Existing Batch File

If your Program Execution Task already has a non-shared batch file and you want to make it shared, click the **Open in editor** button in the batch file editor. This will open the full Script Editor window, where you can mark the batch file as shared and set security options for it.

Changing Shared Batch File References

When you insert a shared batch file call from the batch editor, the editor automatically adds a reference to the shared batch file, which causes adTempus to extract the batch file to a temporary folder at runtime.

If you no longer need to reference a shared batch file, or want to reference a shared batch file without inserting a call to the file using the editor, you can add or remove **Included Batch Files** in the script editor. If you are editing your batch file in the smaller batch file editor in the Program Execution Task properties window, click the **Open in editor** button to display the full editor where this list is available.

Process Termination Task

Process Termination Task

The Process Termination task allows you to have adTempus terminate a process that is executing outside of adTempus.

If the process is running as a service process, use the Service Control Task instead to stop the service cleanly.

Be sure you understand how adTempus terminates a process before using this task. Terminating a process is not always "clean," and this can lead to problems (such as data corruption) with your application.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

Process Condition 272
Process Trigger 240
Service Control Task 204

Reference

Process Termination Task Properties..197

Process Termination Task Properties

The Process Termination Task Properties window is shown when you edit a Job Step that executes a Process Termination Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard response events, this task defines the following events:

Event	Description
No processes found to terminate	Occurs if no matching processes were running.
One or more proccess(es) could not be terminated	Occurs if one or more instance of the process could not be terminated.
One or more process(es) terminated	Occurs if one or more instance of the specified process was terminated.

Process Termination

Process to terminate

Specify the executable file name of the process that adTempus should terminate. You must specify the executable name (e.g., "notepad.exe"). If there is more than one program on the computer with the same executable file name, you may optionally include the full path of the version of the process that you want to monitor.

Note that this must be an executable (.exe). The task cannot target batch files (.bat, .cmd), scripts (.vbs, .js), etc.

If more than one instance of the process is running, all instances will be terminated.

Also terminate any child processes of the target process

Check this option to also terminate any processes that the target process has launched.

Job Variables

The Process Termination Task sets the following Job Variables, which can be used by subsequent operations in the job:

Variable	Description
ProcessTermination.KilledProcessCount	The number of processes matching the target process name that were successfully terminated.
ProcessTermination.FailedToKillProcessCount	The number of processes matching the target process name that were not successfully terminated.
ProcessTermination.KilledChildProcessCount	The number of processes terminated that were child processes of target processes.
ProcessTermination.FailedToKillChildProcessCount	The number of processes not terminated that were child processes of target processes.
ProcessTermination.KilledChildProcesses	Comma-separated list of the names of the child processes that were terminated.
ProcessTermination.FailedToKillChildProcesses	Comma-separated list of the names of the child processes that could not be terminated.

Related Concepts

Process Termination Task 196

Script Execution Task

Script Execution Task

The Script Execution task allows you to run scripts written using the following scripting technologies:

- Microsoft.NET (VB.NET and C#)
- Windows Script Host (VBScript and JScript)
- Windows PowerShell

The script may be stored in adTempus or it may be an external script file.

Although external script files can also be executed using a Program Execution task, executing them using a Script Execution task allows your script to interact with adTempus to write messages to the Job Log, get or set Job Variables, return a result code to adTempus to indicate the success or failure of the script, and more. See the Script Integration topic for more information.

Related Concepts

Script 314
Tasks 147
Steps Overview 145

Related Topics

Script Properties Window 316
Scripts 312
Interacting with adTempus from Scripts 326
Returning a Result From a Script 329
Inline Functions 323
Script Library 320

Reference

Script Execution Task Properties 200

Script Execution Task Properties

The **Script Execution Task Properties** window is shown when you edit a Job Step that executes a Script Execution Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.

★ Event	Description
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard response events, this task defines the following events:

Event	Description
Task completed with an exit code that meets these criteria	Occurs when the target script finishes and returns a result that matches the criteria you specify.
Task executed but returned failure result	Occurs when the target script finishes and, based on the **Success Criteria** on the Options page, is determined to have failed.
Task returned successful result	Occurs when the target script finishes and, based on the **Success Criteria** on the Options page, is determined to have succeeded.
Task will be terminated because it exceeded its allowed execution time	Occurs once the time limit has elapsed, but before adTempus attempts to terminate the process. Respond to this event if you want to use your own script or other approach to end the script.
Task was terminated because it exceeded its allowed execution time	Occurs once the time limit has elapsed, after adTempus has terminated the process.
Script returns a result that meets these criteria	Occurs when the script returns a result that matches (exactly) the string you specify.
Execution time reached a threshold	Occurs when the script runs for longer than the specified time.

Script

Script to Execute

Your script may be either an internal script (stored within adTempus and optionally shared with other jobs) or an external script (an external script file). The way the script is executed, and the options available to it, are the same either way.

Execute a script stored in adTempus

Select an existing script or create a new one. See the Scripts topic for information on working with internal scripts.

Execute a script stored in an external file

Specify the script file to execute. You must also specify the **Language** of the script. If you select a language other than one of those listed, the language name you enter must be a scripting language recognized by the Window Scripting Host on the adTempus server.

Run isolated from other scripts

When this option is checked, this script will run in a host process that is not shared by any other scripts. See the Script Security and Isolation topic for more information. If this option is not checked,

- For a "script stored in adTempus," the setting from the Script Properties is used.
- For a "script stored in an external file," the script is not run in isolation.

PowerShell Options

If you select a PowerShell script, the following additional options are available.

Capture console output from script

Check this option to capture the output written by the PowerShell script to stdout using `write-output`.

Command-Line Parameters

Use this box to pass command-line parameters to the script. Parameters can be retrieved using the $args collection in the script.

Options

Success Criteria

The Success Criteria determine whether the step is reported as Successful or Failed. Three options are available:

- **The script will return a numeric result code.** Your script will return a numeric result (see Returning a Result From a Script). The step will be reported as successful if the result meets the criterion you specify. By default, adTempus expects your script to return a value of "0" if it succeeds, consistent with the convention used by most programs.
- **Report the step as successful unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task was successful. You can change the status of the step using a Job Control Action.
- **Report the step as Failed unless instructed otherwise by a Response.** adTempus will ignore the exit code and assume that the task failed. You can change the status of the step using a Job Control Action.

You can also test for any number of specific exit codes or string results and take action based on them using Responses for the step. For example, you may want to report the step as failed if the exit code is greater than 0, but send e-mail notification to an administrator if the exit code is greater than 128.

Time limit

Specify the maximum time (in minutes) that the script should be allowed to run. This feature protects against "runaway" scripts.

If you are running an internal script, adTempus will use the lesser of the value you specify here and the value specified on the Script's properties.

Related Concepts

Script Execution Task 199

Service Control Task

Service Control Task

The Service Control task allows adTempus to start, stop, and monitor Windows service applications.

When you configure adTempus to monitor a service, adTempus can detect when the service is stopped (e.g., due to an application failure) and restart the service automatically.

Like all other adTempus tasks, the Service Control Task is executed in the security context of the user account specified for the job. Therefore that user must have the authority to control the service you select, or the task will fail. To avoid running the job under an Administrator account, you can check the Use system context for certain operations option for the Credential Profile that is used for the job.

Using adTempus to Automatically Restart a Failed Service

To automatically detect and restart a failed service, configure the task as follows:

1. Select the service you wish to monitor and set the **Control Type** to "Start the service."
2. Check the **Monitor the service** option.
3. On the Responses page create a new response.
4. Add the "Service failed or was stopped outside of adTempus" event to the response.
5. Add a "Control a job or job step" (Job Control) action to the response. Set the Job Control action's **Action to take** to "Restart the step."

6. Optionally (but recommended) specify a retry limit for the action so that adTempus does not end up continuously restarting a service that fails repeatedly.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

Process Termination Task 196
Process Condition 272
Process Trigger 240

Reference

Service Control Task Properties 205

Service Control Task Properties

The **Service Control Task Properties** window contains the settings for a job step that executes a Service Control Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.

★Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

⭐In addition to the standard Response Events, the Service Control Task defines the following special events:

Event	Description
Task returns a failure result	Occurs if adTempus is unable to start or stop the target service.
Task returns a successful result	Occurs if adTempus succeeds in starting or stopping the service. **Note:**This event occurs before adTempus begins monitoring the service, so you can use it to respond to the service start without waiting for the monitoring to complete.
Service failed or was stopped outside of adTempus	Occurs if adTempus detects that the service is no longer running. Valid only if the Monitor the service option has been selected.

Service Control

Service Location

Specify whether the target service is on the adTempus server or a different computer.

Service

Select the service you wish to control. The list of services is taken from the adTempus server.

You may also type in the name of the service if necessary. For example, the service may exist on the Agents where the job will run, but not on the Master computer where the job is maintained. Since the service list would come from the Master, in this case you would have to type in the service name.

If you enter the service name yourself, you must enter the "service name," which is often not the same as the "display name" that appears in the Service Control Manager. The service name is the same name you use when controlling the service through the "net start" and "net stop" commands. If you do not know the service name, look in the Registry under key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`. This key contains a subkey for each service; the key name is the service name.

Control Type

Select the action you want adTempus to perform:

- **Start the service.** adTempus will start the specified service. If the service is already started, adTempus will log an informational message but the step will succeed.

  Optionally, specify any **Startup Parameters** that should be passed to the service.

- **Stop the service.** adTempus will stop the specified service. If the service is not running, adTempus will log an informational message but the step will succeed.

Options

When **Start the service** is selected as the control type, the following additional options are available:

Monitor the service

When this option is selected, adTempus will monitor the service. If adTempus detects that the service has stopped, it will fire the "Service Failed" event. By creating a Response to this event you can have adTempus restart the service (by restarting the step) or take other actions as appropriate.

When this option is selected, the job step does not end until the service has stopped (and no responses cause it to restart). Therefore, execution of the job will not continue past this step until the service stops. If you use this option you generally will want this step to be the only—or last—step of the job.

adTempus can only detect that the target service is no longer running. It cannot determine *why* the service is no longer running. Therefore any of the following circumstances could trigger the task's failure response:

- The service application aborts ("crashes") in some way.
- A user, or some other application, stops the service (e.g., using the Service Control Manager).
- The operating system stops the service because the computer is shutting down. You should therefore make sure that adTempus is the first service to be stopped when the computer is being shut down or restarted.

Stop the service if the job is terminated

When this option is selected, adTempus will stop the service if the job is terminated (either through a manual abort or as a result of a Job Control action). The service is **not** stopped if the adTempus service is stopped.

This option is only available if the **Monitor the service** option is selected.

Related Concepts

Service Control Task..204

Web Request Task

Web Request Task

The Web Request task allows adTempus to request a page or file from a Web server or call a method on a Web service.

This task can be used to simply "ping" a URL on the server (for example, if the Web application has offline processing that is triggered by requesting a special URL on the server) or to download a page or file from the Web server.

Pages and files retrieved by the task can be saved in the job's history and/or written to a specified location on the adTempus server.

The Web Request Task is used to request a URL as part of job processing. If you want to trigger a job based on the response (or non-response) of a Web server, use the Computer Monitor Trigger instead.

To download files using FTP or SFTP, use the File Transfer Task.

Related Concepts

Tasks 147
Steps Overview 145

Related Topics

Computer Monitor Trigger 213
File Transfer Task 162

Reference

Web Request Task Properties Window 209

Web Request Task Properties Window

The **Web Request Task Properties** window is displayed when you edit a job step that runs a Web Request Task.

Property Pages

Property pages common to all Tasks

General

Name for this step (optional)

Optionally, specify a descriptive name for the step.

Enable this step

Uncheck this box to disable the step. If the step is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this step.

Conditions

The **Conditions** page defines conditions that must be satisfied before the step is run.

Condition Criteria

The Condition Criteria determine how the conditions should be evaluated:

- Execute only if all conditions are met. The step is only executed if all of the listed conditions are met.
- Execute if any condition is met. The step is executed if any of the listed conditions is met.

If Conditions are not satisfied

The satisfaction options determine what adTempus should do if the conditions are not satisfied:

- Fail the step. The step is not executed; the status is set to Failed.
- Execute anyway. The step is executed anyway.
- Skip the step (do not report as a failure). The step is not run, but it is not treated as a failure. The status of the step is reported as "Skipped (conditions not met)."

Conditions List

The **Conditions** list lists the conditions that have been defined for the step. You can add, edit, or delete conditions. See the Conditions topic for information on the available condition types.

Variables

The **Variables** page allows you to define Job Variables for this step. You can add new Variables, or override the values of Variables inherited from the Job. Any Variables you define or override here affect only this step of the Job.

To set Variables that apply to the entire job, use the Variables page in the Job properties.

Responses

The **Responses** page defines the actions that adTempus should take in response to events that are fired during execution of the step. You can add, edit, delete, or reorder responses.

All job steps support the events listed below. Some tasks may define additional events .

Event	Description
Step Started	Occurs at the beginning of the step.
Step Ended	Occurs at the end of the step, regardless of the step result.
Step Failed	Occurs if the step fails for any reason.
Step Restarted	Occurs if the step is restarted due to a Response.
Restart Limit Exceeded	Occurs if the restart limit is exceeded.

Event	Description
Step skipped	Occurs when the step is skipped, either because conditions were not met (if the **Skip...** option is selected on the Conditions page) or due to a skip option specific to the task.
One or more conditions failed	Occurs if one or more Conditions for the step is not satisfied.
Conditions not met within the specified time	Occurs if the job or a step within the job has been waiting for conditions for longer than the specified time.

In addition to the standard response events, this task defines the following events:

Event	Description
The Web request was executed (regardless of the outcome)	Occurs when the Web request has been executed, regardless of the result.
The HTTP request returns the following status code	Occurs if the HTTP request returns the specified status code. For example, a status code of 404 is returned if the page or file is not found.

Web Request

URL to Request

Enter the URL of the Web page or file to request. For example, `http://www.example.com`. The URL may contain Job Variables.

Credentials

Specify the credentials, if any, needed for this Web resource.

- If the Web resource does not require credentials (i.e., it accepts anonymous requests), select "Use the job's credentials."
- If the Web server is configured to use Windows authentication, and the account that the job is running under has the necessary permission on the Web server, select "Use the job's credentials."
- Otherwise, you must select "Send the following credentials to the Web server" and provide the necessary credentials. Type the user name in the **User ID** box. If this is the first time this user ID has been used for a Web-based task in adTempus, you will be prompted to complete a Credential Profile for the user ID, specifying the password.

Save the page or file returned by the server

Use these options if you want to save the page or file returned by the Web server. You can choose to save the file in the job's history and/or save the file to a specified directory.

Job Variables

The Web Request Task sets the following Job Variables, which can be used by subsequent operations in the job:

Variable	Description
WebRequest.RequestURL	The URL requested by the task.
WebRequest.StatusCode	The HTTP status code returned by the remote server.
WebRequest.ResponseFileName	The path/name of the file where the response is saved (if the option to save the server response to a file has been specified for the step).

Related Concepts

Web Request Task 208

Triggers

Triggers

Triggers determine when a job should be run. Each job may have any number of triggers, allowing it to respond to various kinds of events.

The triggers for a job are managed through the Trigger page of the job's properties.

Available Triggers

adTempus supports the following triggers:

- The Schedule Trigger is used to execute jobs at specific dates and times.
- The Startup Trigger is used to execute jobs whenever the adTempus service is started.
- The File Trigger is used to execute jobs based on file creation, deletion, or modification.
- The Computer Monitor Trigger is used to monitor remote computers and execute jobs when they fail.
- The Event Log Monitor is used to monitor the computer's Event Log and execute jobs when certain events are logged.
- The Process Trigger is used to monitor external processes and execute jobs when they start, end, or exceed memory limits.
- The WMI Trigger is used to monitor Windows Management Instrumentation (WMI) events and trigger jobs based on those events.
- The E-Mail Trigger is used to execute jobs based on e-mail messages that match selection

criteria.

- The Job Trigger is used to trigger a job based on the status of one or more other jobs.

Related Concepts

Key Concepts 81

Related Topics

Comparison of Triggers and Conditions 87
Triggers Page 138

Computer Monitor Trigger

Computer Monitor Trigger

The Computer Monitor Trigger allows adTempus to monitor remote computers and trigger a job when the remote computer fails.

The Computer Monitor can use a simple network ping to verify that the computer is alive and reachable, or it can make an HTTP request to the server and evaluate the result. This feature can be used, for example, to verify that your Web application is running properly.

If you need to request and save a Web page or file as part of your job's processing after the job is triggered by some other mechanism, use the Web Request Task instead.

Related Concepts

Triggers 212

Related Topics

Comparison of Triggers and Conditions 87
Triggers Page 138
Using to Monitor Your Web Application 218
Web Request Task 208

Reference

Computer Monitor Properties 214

Computer Monitor Properties

The **Computer Monitor Properties** window contains the settings for a Computer Monitor Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

Target

Target

If the **Test Method** is "Simple Ping," specify the name or IP address of the computer to ping.

If the **Test Method** is "HTTP Request," specify the URL that adTempus should request. For example:

```
http://10.10.10.201/testapp.aspx
```

The **Target** may contain Job Variables, which will be expanded to resolve the target address.

Test Method

Select the type of test you want to perform:

- **Simple Ping.** adTempus will ping the computer at the interval you specify. If a response is received, adTempus considers the computer to be alive. If no response is received, adTempus considers the computer to be dead, and triggers the job. Note that adTempus cannot distinguish between a failed computer and a failed network connection. Use the "HTTP Request" method for a better evaluation tool.

- **HTTP Request.** adTempus requests a specific page from a Web server, and then evaluates the response it receives to decide whether to trigger the job. See Using adTempus to Monitor Your Web Application for more information.

Response Evaluation

These options apply only when you have selected the "HTTP Request" test method, and determine how adTempus should evaluate the reply it receives from the target computer.

- **Trigger if response matches.** If the response matches the specified text, the job will be triggered.

- **Trigger if response does not match.** If the response does not match the specified text, the job will be triggered.

- **Use a script to evaluate the response.** adTempus will run the script you specify. Information about the response will be passed to the script using the Parameters object and the Job Variables collection. Your script then evaluates the response and returns True if the response represents a failure or False if it represents success.

When you specify text for adTempus to match, adTempus looks for that text anywhere within the response it receives. For example, if you specify that the job should trigger if the response does not match `Application Status: OK`, adTempus will look for the string `Application Status: OK` anywhere within the response, and trigger if it does not find it.

Use regular expressions

If this option is checked, the match text you enter is treated as a regular expression. If a match for that regular expression is found anywhere within the event message, the condition is satisfied.

Test Interval

These options determine how often adTempus will test the server:

- Test every X minutes

- Wait for a reply for Y seconds

- Trigger after Z minutes

The options work together as follows:

- Every X minutes, adTempus will send a ping or HTTP request to the target computer
- adTempus will wait up to Y seconds to receive a response from the target.
  - If no response is received, it considers the target to be dead.
  - If a response is received, it checks the other criteria to see if the target should be considered alive.
- If the target is dead, adTempus will continue checking every X minutes until Z minutes have elapsed since the first failure. If no valid response has been received after Z minutes, the job will be triggered.

If you want the job to be triggered the first time a failure is detected, set Z to be <= X.

Trigger at test interval until successful response received

If this option is checked, after adTempus first triggers the job it will continue to trigger the job at the test interval until the target is found to be alive.

If this option is not checked, adTempus will trigger the job only once per failure (that is, it will not trigger the job again unless the target is restored and then fails again).

Trigger again when successful response received

If this option is checked, adTempus will trigger the job again when a successful response is received from the target.

This option is useful if you are using the trigger to send a notification message to someone when the target fails. If this option is checked, a message would be sent when the target fails, and another sent when the target is restored.

The trigger sets a Job Variable named "ComputerMonitor.Action" (see below) that indicates whether the job is being triggered because the target failed or because it was restored. You can test this variable using a Script Condition to determine which steps of your job to run depending on the action.

For example, Step 1 could use a Script Condition so that it only executes if the Action variable is `ConnectionFailed`. Step 1 therefore would be run when the target computer fails.

Step 2 could use a Script Condition so that it only executes if the Action variable is `ConnectionRestored`. Step 2 therefore would be run when the target computer was restored.

Additional Options

Identity

Specify the credentials that will be used for the connection to the target computer.

- If you are using a simple ping test, or requesting a page from a Web site that does not require authentication, check the **The connection can be made without authentication...** option.

- If you are requesting a page from a Web site that requires authentication,
  - Check the **The connection can be made without authentication...** option if the Web server accepts Windows authentication and the user whose account is being used to run the job has permission to request the target Web page.
  - Otherwise, you need to specify a user ID and password for adTempus to use when connecting to the target server. Enter the user ID to select an existing Credential Profile or to create a new one.

Job Variables

The Computer Monitor Trigger sets the following Job Variables, which can be used by a selection script or by other scripts or notification messages in the job.

Parameter Name	Description
ComputerMonitor.ConnectionStatusCode	A code indicating the result of the connection attempt.
ComputerMonitor.Action	A string indicating whether the trigger is firing because the connection has been broken or because it has been restored: • "ConnectionFailed": No connection could be made to the server, or it returned an invalid response. • "ConnectionRestored": The connection has been restored, or the server is now returning a good response.
ComputerMonitor.FailureCount	If ComputerMonitor.Action is `ConnectionFailed`, FailureCount Indicates the number of consecutive unsuccessful responses that have been received since the last successful response. If ComputerMonitor.Action is `ConnectionRestored`, FailureCount Indicates the number of consecutive unsuccessful responses that were received before the connection was restored. You can use this parameter to escalate your responses. For example, you could configure your job to send a warning message to certain users after the connection has been down for 5 minutes, and send a more urgent message if the connection is still down after 20 minutes.
ComputerMonitor.Target	The Target that the trigger is trying to connect to.
ComputerMonitor.ErrorMessage	The most recent error message for the

Parameter Name	Description
	connection.
ComputerMonitor.LastSuccessTime	The date/time (in UTC) of the last success response from the server.

Related Concepts

Computer Monitor Trigger 213

Related Topics

Using to Monitor Your Web Application 218

Using to Monitor Your Web Application

A simple ping can tell you whether a remote computer is running and reachable, but this often does not provide you with enough information about the computer.

Just because the computer is "alive," for example, doesn't mean that the Web server software is running properly.

adTempus provides additional monitoring flexibility through its ability to make an HTTP request to a remote computer, and then evaluate the response that the computer returns.

A Simple Test

In a simple situation, you may only want to verify that Internet Information Server is running and serving pages properly. To do this, you would create a simple ASP page on the Web server, called, for example, "testapp.asp." It might look like this:

```
<%@ Language=VBScript%>
Application Status: OK
```

You would then configure the Computer Monitor to request this page, and to trigger if the response did not match `Application Status: OK`.

Note that we did not format the output from this ASP page as a properly-formed HTML page. Since it's not intended for a browser, there's no need to do this. In fact, the HTML tags would make the error message harder to read if you include it in notification messages.

Doing More

The ASP page or other scripted page you have adTempus request can do as much processing as you need and then return an appropriate response. For example, it could verify that your web server is able to connect to your web server, check that disk space is sufficient, etc.

Using the Results

The computer monitor sets two Job Variables that you can use when assessing the state of the target computer.

The **ConnectionStatusCode** parameter indicates whether adTempus actually received a response from the computer.

- If the **ConnectionStatusCode** is 0, adTempus was able to retrieve the requested page, and the **ResponseText** parameter contains the complete text returned by the server.
- If the **ConnectionStatusCode** is not 0, this indicates a network or server error that prevented adTempus from receiving a reply. For example, the target computer may be unreachable due to a network problem. In this case, the **ResponseText** parameter contains an error message describing the problem.

Using the **ResponseText** Script Parameter, you can execute certain steps of your job depending on the nature of the error returned, and include the error message from the server in notification messages that get sent by adTempus.

Related Concepts

Computer Monitor Trigger 213

Reference

Computer Monitor Properties 214

Event Log Monitor Trigger

Event Log Monitor Trigger

The **Event Log Monitor Trigger** allows adTempus to monitor the Event Log on the host computer and trigger jobs when selected events are logged.

For example, you could have a job that is run whenever a failed logon attempt is reported by the operating system, or a job that is run whenever SQL Server reports a replication error.

Each time adTempus is started, it reviews any events that were logged while it was stopped, and triggers based on those events as appropriate. It then begins watching for new events and triggering based on them.

Related Concepts

Triggers 212

Related Topics

Comparison of Triggers and Conditions 87
Triggers Page 138

Reference

Event Log Monitor Properties 220

Event Log Monitor Properties

The **Event Log Monitor Properties** window contains the settings for an Event Log Monitor Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

Event Selection

This page defines the conditions for triggering a job. adTempus will trigger your job each time a new event is reported that meets all of the criteria you specify.

Log

Select the log to monitor. A trigger can only monitor a single log.

Sources

Select the event sources (applications) whose events you want to check. You may select any number of sources; if you do not select any source, all sources will be considered.

For example, you might want to trigger based on Error messages logged by a particular application in the Application log. Or you might want to trigger any time an Error message is logged in the System log.

Category

This option is available only if you have selected a single event source. Select the category of messages you want adTempus to examine.

Types

Select the types (severities) to consider. If none are selected, all types are considered.

Include Event IDs

If you want to trigger only when specific events are logged, enter the IDs of those events here. You can specify as many events as you wish, separated by commas.

Event IDs are specific to each application (Event Source). To determine which events IDs you want to capture, consult the documentation for the application in question, or use the Windows Event Log Viewer to examine events the application has logged. The Event Log Viewer will show you the Event IDs for those events.

If you have selected multiple Event Sources, you should not specify Event IDs unless you are sure you know what you are doing (i.e., all of the sources you have selected use the same Event IDs to mean the same thing).

Exclude Event IDs

If you want to ignore certain events, enter the IDs of those events here. You can specify as many events as you wish, separated by commas.

For example, you might want to trigger any time an event of type Error is logged by Source `My Application`, except if the event ID is `117`.

Select events whose message matches

Check this option to select events based on the text of the event message. If the message contains the text you specify, it will be selected. The message does not have to match the specified text exactly: it may contain the specified text anywhere within it.

Use regular expressions

If this option is checked, the match text you enter is treated as a regular expression. If a match for that regular expression is found anywhere within the event message, the condition is satisfied.

Use a script to select events

adTempus will run the script you specify. Information about the event will be passed to the script using the Parameters object and through Job Variables (see list below). Your script then evaluates the event and returns **True** if adTempus should trigger the job, or **False** if it should not.

adTempus calls the script only for events that have satisfied all of the other criteria for the trigger.

Job Variables

The Event Log Monitor Trigger sets the following Job Variables, which can be used by a selection script or by other scripts or notification messages in the job.

Parameter Name	Description
EventLogMonitor.EventLog	The name of the log in which the event was recorded (e.g., "Application").
EventLogMonitor.EventSource	The name of the event source. This is the name of the source as it appears in the Registry, and may not be the same as the name displayed in the user interface.
EventLogMonitor.EventCategory	The numeric event category.
EventLogMonitor.EventCategoryName	The name of the event category.
EventLogMonitor.EventType	The numeric event type (severity).
EventLogMonitor.EventTypeName	The event type name ("Error", "Warning", "Information")
EventLogMonitor.EventID	The numeric event ID.
EventLogMonitor.EventMessage	The full text of the message.
EventLogMonitor.EventTimestamp	The timestamp for the message in format "yyyy-MM-dd HH:mm:ss".
EventLogMonitor.EventKeywords	A comma-separated list of keywords for the event.

Related Concepts

Event Log Monitor Trigger......219

E-Mail Trigger

E-Mail Trigger

The **E-Mail** **Trigger** allows you to retrieve e-mail messages from POP3 and IMAP servers and trigger a job if messages match filtering rules (based on properties such as message contents, message headers, and attachment names). You can also save the messages and/or message attachments to disk for processing.

The E-Mail Processing Task and the E-Mail Trigger both offer exactly the same functionality, except that the E-Mail Trigger triggers a job when it finds messages that match the selection criteria, while the E-Mail Processing Task must be included in a job that is triggered through some other mechanism.

For example, if you want to run a job each time an e-mail message arrives, you would use the E-Mail Trigger. If instead you wanted to run a job at 6:00PM to process any messages that have arrived during the day, you would use a Schedule Trigger to run the job, and an E-Mail Processing Task to download the messages.

Related Concepts

Triggers 212

Related Topics

E-Mail Processing Task 157
Comparison of Triggers and Conditions 87
Triggers Page 138

Reference

E-Mail Trigger Properties 223
E-Mail Selection Filter 227

E-Mail Trigger Properties

The **E-Mail Trigger Properties** window contains the settings for an E-Mail Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

Message Source

Mail connection to use

Select an existing Incoming E-Mail Provider or create a new one. The e-mail provider represents a connection to a POP3 or IMAP mail server using a single login account.

IMAP Folder

For an IMAP connection you may optionally enter the name of the folder to select messages from. Leave blank to select from the default Inbox folder for the account.

Delete selected messages

If this option is checked, adTempus will delete from the mail server each message that is selected and processed by the e-mail processing operation. Messages that do not meet the selection criteria are not deleted.

If this option is not checked, adTempus will leave the messages on the server. However, adTempus remembers which messages it has already seen and will not reprocess the messages again in the future unless you clear the message list as described in the next section.

Some mail servers (such as Gmail) may not delete messages even if adTempus tells them to.

Clear list of previously-processed messages

Click this button to clear the list of messages that adTempus has already examined in this mailbox. This will cause adTempus to examine all messages in the mailbox the next time it connects to the server. Use this option if, for example, you have changed the message filtering rules and want adTempus to re-examine messages to select messages that were not processed in the past because they did not meet the selection rules.

Check for new messages every

Specify the interval (in minutes) at which adTempus should connect to the mail server to check for new messages.

Selection Criteria

This page allows you to define criteria controlling which messages adTempus will select for processing. If no criteria are defined, adTempus will select all new messages in the mailbox.

adTempus remembers which messages it has already seen, and will never re-read messages it has already seen, unless you clear its list using the **Clear list of previously-processed messages** button.

Selection Rules

If no rules are defined, adTempus will select all messages in the mailbox. To filter messages based on criteria based on their subject, content, sender, attachments, etc., add one or more selection filters.

If you add more than one rule, specify whether adTempus should select messages that match **Any** rule or only messages that match **All** rules.

Selection Script

For more advanced filtering, you can use a .NET script to filter the message. For each message that meets the basic selection criteria (if any), adTempus will run your script. Your script has access to the Parameters.Message object, which exposes the message and its attachments through a MailMessage object.

Your script must return True if adTempus should select the message or False if it should not.

Trigger separately for each message

If this option is checked adTempus will start a new instance of the job for each message that matches the selection criteria.

If this option is not checked, adTempus will start a single instance of the job even if it found more than on message that matches the selection criteria.

Message Saving

The options on the Message Saving page allow you to save selected messages and/or their attachments to disk.

Save format

Select the format to save the message in:

- **RFC 822.** The message is saved in standard RFC 822 format with file name `message.eml`. Any attachments are included in the file as embedded MIME sections.
- **RFC 822 with attachments.** The message is saved in standard RFC 822 format with file name `message.eml`, and attachments are also saved in separate files.

- **HTML with attachments.** The body of the message is saved to an HTML file named `message.htm`. Attachments are saved in separate files.
- **Text with attachments.** The body of the message is saved to a text file named `message.txt`. Attachments are saved in separate files.
- **Attachments only.** The body of the message is not saved, but all attachments are saved in separate files.

Save messages to folder

Specify the base folder where messages should be saved. adTempus will create a uniquely-named folder under this folder for each message. The folder will contain a file for the message body (named as specified above based on the **Save format**) and/or the attachments for the message.

adTempus will store a list of the folders that it creates in the **EMailProcessing.MessageLocation** Job Variable.

For example, you set the save location to `c:\messagedownload`. adTempus processes 2 messages and saves them in format **HTML with attachments**. The variable will be set to:

```
c:\messagedownload\{C7C9286D-607F-4192-9960-
EDED8624A4BE};c:\messagedownload\{837FC8D4-314D-41E3-B95E-412CB84694E5}
```

The messages can be found in files `c:\messagedownload\{C7C9286D-607F-4192-9960-EDED8624A4BE}\message.htm` and `c:\messagedownload\{837FC8D4-314D-41E3-B95E-412CB84694E5}.htm`.

The attachments for each message can be found in the corresponding folders.

Job Variables

This trigger creates the following Job Variables, which are available for the remainder of the job:

Name	Description
EmailTrigger.MessageId	The message ID (assigned by the e-mail server) for the selected message. This variable is set only if the Trigger separately for each message option is checked.
EmailTrigger.MessageSubject	The subject of the selected e-mail message. This variable is set only if the Trigger separately for each message option is checked.
EMailTrigger.MessageLocation	A semicolon-delimited list of the files or directories created if messages or attachments were saved. See the Message Saving section for more information.
EmailTrigger.SelectedMessageCount	The number of messages selected by trigger (the number of messages that meet the selection criteria).

Related Concepts

E-Mail Trigger 222

Reference

E-Mail Selection Filter 227

E-Mail Selection Filter

The E-Mail Selection Filter window defines a selection rule for processing e-mail messages with an E-Mail Trigger or E-Mail Processing Task.

Settings

Filter based on

Choose whether you will filter based on properties of the message (such as sender, subject, etc.) or of the message's attachments (file name, content, etc.)

Filter on property

Select the message or attachment property to search.

Message properties

- **From Address.** Searches the sender e-mail address (address part only; not the name)
- **To Address.** Searches all TO e-mail address (address part only; not the name)
- **CC Address.** Searches all CC e-mail address (address part only; not the name)
- **To or CC Address.** Searches all TO and CC e-mail addresses (address part only; not the name)
- **Subject.** Searches the message subject
- **Message Body.** Searches the message body (the message text)
- **Message Header.** Searches in the message header with the specified name.

Attachment properties

- **File Name.** Searches the names of all attachments to the message
- **File Extension.** Searches the file extensions of all attachments to the message. When entering the value to match, the "." at the beginning of the extension is optional (`pdf` and `.pdf` are equivalent)
- **MIME Content Type.** Searches the MIME content type of all attachments to the message.
- **File Content.** Searches the contents of all attachments.

Matching rule

Select the matching rule to apply:

- **Contains.** The selected property contains the specified value
- **Does not contain.** The selected property does not contain the specified value
- **Equals.** The selected property matches the specified value fully (no other text appears before or after the value)
- **Does not equal.** The selected property does not match the specified value fully (the value does not appear in the property, or appears with other text)
- **Starts with.** The selected property starts with the specified value

Value to match

Specify the text to search for. Job Variables can be used (variable tokens are expanded before matching). Matching is always case-insensitive ("This Value" will match "this value").

Use Regular Expressions

Check this option to use a Regular Expression for pattern matching. Use the **Test Regular Expression** button to test your Regular Expression and make sure it matches as you expect.

Examples

Select all messages that have one ore more PDF attachments: **Attachment Properties/File Extension/Equals** `pdf`

Select all messages from "someone@example.com": **Message Properties/From Address/Equals** `someone@example.com`

Related Concepts

E-Mail Trigger 222

Reference

E-Mail Trigger Properties 223

File Trigger

File Trigger

A **File Trigger** is a trigger that causes a job to be executed whenever specified files are created, modified, or deleted.

Use the File Condition instead to make execution of a job or step conditional on the presence or absence of a file, when the job is triggered by a different kind of trigger.

When you create a File Trigger, you specify the files or directories that adTempus should watch for, and under which conditions the job should be triggered. Additional options can be

⭐used to prevent adTempus from triggering for a file while another application is still updating it.

The File Trigger works with local drives, network drives, and FTP and SFTP servers.

The File Trigger works by periodically scanning the directories being watched by the trigger and comparing the current state to the previous state. Options for the trigger allow you to specify how often adTempus should look for changes. Note that the trigger does not respond to individual file events. Therefore if a file is deleted and re-created in between two scans of the folder, adTempus will see this as a file modification, not as a file deletion and a file creation.

Changes Made While adTempus is Not Running

adTempus saves the current state of the directories it is monitoring each time it makes an evaluation pass. This state information is stored in the adTempus database, and so is not lost if the adTempus service is shut down. Therefore when adTempus starts, it compares the current file state to the state information it saved before it shut down, and triggers jobs based on the differences. That is, adTempus will respond to files that were added, changed, or deleted while adTempus was not running.

Removable Drives and Network Drives

If the target files are on a removable drive or a network drive and the drive is removed or disconnected, adTempus will ignore those target files until the drive is reattached. It will then compare the new state of the files on the drive to the last snapshot it took before the drive was disconnected, and assess file creations, deletions, and modifications accordingly.

In adTempus 3 and earlier, adTempus acted in this scenario as though the files had been deleted when the drive was disconnected, and created again when the drive was re-connected.

The File Trigger cannot be used to monitor network drives mapped to a local drive letter. You must use the UNC path to the files rather than the drive letter. This is because drive letters mapped outside adTempus are not available to adTempus, and drive mappings you establish on the Resources page are not connected until after the job has been triggered. See the "Network Access for Jobs " on page 482 topic for more information.

Related Concepts

Triggers 212

Related Topics

File Operation Tasks 161
File Condition 260
Comparison of Triggers and Conditions 87
Triggers Page 138

Reference

File Trigger Properties 230
File Specification Properties 235

File Trigger Properties

The **File Trigger Properties** window allows you to view or modify the settings for a File Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

File Selection

File Location

If the files you want to monitor are located on a local drive (a drive attached to the computer where the adTempus service is running) or on a network share (using a UNC path), choose "Local or Network File" as the File Location.

adTempus can also monitor files on a remote computer using an FTP or SFTP connection. To use this feature, select an existing File Service Provider or click the Add button to create a new one to provide the settings adTempus needs to connect to the server.

Files

Specify the files or directories to watch. You may specify a single file (e.g., "c:\data\jobtrigger.txt") or use wildcards (e.g., "c:\datafiles\newfiles\*.dat").

You may add as many file specifications as you wish. The job will be triggered when a file is found that matches any of the file specifications you have provided and matches all the other conditions you have specified.

The File Trigger cannot be used to monitor network drives mapped to a local drive letter. You must use the UNC path to the files rather than the drive letter. This is because drive letters mapped outside adTempus are not available to adTempus, and drive mappings you establish on the Resources page are not connected until after the job has been triggered. See the "Network Access for Jobs " on page 482 topic for more information.

Polling Interval

Specify the interval at which adTempus will check for new, deleted, and modified files. At each polling interval, adTempus will take a snapshot of the target directory and compare it to the previous snapshot to determine which files have been created, deleted, or changed.

Set the polling interval to a smaller value if you need to respond quickly to file system changes, or a larger value if reaction time is not important.

Trigger Conditions

Select the conditions under which the job will be executed. You must select at least one option.

- **Trigger when file is created.** adTempus will trigger the job when it finds a file that was not present in the previous snapshot. Note that if a file is deleted and re-created within the Polling Interval, this will be seen by adTempus as a file modification, not as a deletion and a creation. This is because the trigger works by scanning the folder and comparing the contents between scans; it does not respond directly to individual file events. Therefore you should generally also use the option to trigger when the file is modified, unless your trigger files will have unique names.

- **Trigger when file is modified.** adTempus will trigger the job when it finds a file with a newer timestamp than in the previous snapshot. (If the file did not exist in the previous snapshot, adTempus will not trigger the job unless the "create" option is also checked.)

- **Trigger when file is deleted.** adTempus will trigger the job when a file that was in the previous snapshot is not found in the current snapshot. (The file has not necessarily been deleted; it could have been renamed or moved to a different folder.)

If the target files are located on a network drive or removable drive, adTempus will ignore the files if the drive is removed or disconnected. More information.

When you use the **Trigger when file is modified** option, be sure the **Minimum Interval** and/or **Polling Interval** options are set to appropriate values to avoid triggering the job continuously.

For example, suppose you are monitoring a log file for changes. The first time adTempus detects a change to the file, it will trigger your job. If the application that is writing to the log file continues to write to it, adTempus will continue to trigger your job each time it detects a change, so it will continue to trigger each time the Polling Interval elapses. If your polling interval is 5 seconds, adTempus will fire the job every 5 seconds.

Use the Wait for exclusive access and Wait until... options (on the **Additional Options** page) to prevent adTempus from triggering for a file that another application is still updating.

Trigger Mode

The Trigger Mode determines whether adTempus will start a single instance of the job, or multiple instances.

Each time the File Trigger scans the directories, it produces a list of all the files that meet the trigger criteria. Because adTempus makes this evaluation at regular intervals (rather than responding instantly to each change in the file system), it is possible for more than one file to meet the trigger criteria.

For example, suppose you have a trigger configured to fire any time files are added to a particular directory. In between evaluations of the directory, five new files are copied into the directory. On the next evaluation, adTempus will find those five files, and **Trigger Mode** setting determines how it will handle the scenario.

Trigger a single instance as soon as any matching file is found

If the **Trigger Mode** is set to "Trigger a single instance as soon as any matching file is found," adTempus will start a single instance of the job, even if more than one file has been found. The FileName job variable will contain the names of all the files found on the current evaluation pass.

Typically you would use this option if your program is being called to process an entire directory. For example, your program already has logic to scan an entire folder and process all the files in it; you only need adTempus to start it when there are files there to process.

In this scenario, you need to make sure you have configured the job so that it will not be triggered too often. For example, set the job's Multiple Instances rule to prevent duplicate instances of the job from being started at the same time. Otherwise, adTempus may trigger your program again while it is still processing files from the first trigger.

Include all matching files in FileName variable

When this option is checked, adTempus will still evaluate all files for matches, and the "FileName" Job Variable set by the trigger will contain a semicolon-delimited list of all matchin files. If this option is not checked, adTempus stops evaluating files as soon as it finds a matching file, and the FileName variable contains only the name of that file.

Trigger a separate instance for each matching file

If the **Trigger Mode** is set to "Trigger a separate instance for each matching file," adTempus will start a separate instance of the job for each of the matching files it has found, and each instance will have the FileName job variable set to the name of the file for which it is being executed.

Typically you would use this option if your program is being called to process files individually. For example, you need to run a data processing utility on each file that is uploaded into a directory. When you configure the Program Execution Task to run your program, you can include the token "%FileName%" in the command-line parameters, and adTempus will replace "%FileName%" with the name of the triggering file.

If the trigger is monitoring a directory and 5 files are added to it at once, adTempus will start 5 separate instances of the job (all running at the same time), each one linked to one of the files.

Trigger a single instance once all file specifications have been met

If the **Trigger Mode** is set to "Trigger a single instance once all file specifications have been met," adTempus waits until *all* of the file specification rules are satisfied, and then triggers a single instance of the job. For example, if you have one File Specification that is set to wait for `c:\inputs\file1.txt` and a second File Specification that is set to wait for `c:\chained results\processing results.log`, then the job will not be started until both of those files are created.

Minimum Time Between Triggers

Specify the minimum time (in seconds) that must elapse after this trigger fires before adTempus will fire it again.

adTempus evaluates the files approximately every 10 seconds; specify a longer time if appropriate.

This option is not available if you have selected the **Start a separate instance for each file**option. When that option is checked, adTempus will always start a new instance of the job for each file that meets the trigger criteria.

You should use this option and/or the options available on the job to prevent duplicate instances to prevent your job from being triggered continually.

For example, suppose you are monitoring a log file for changes. The first time adTempus detects a change to the file, it will trigger your job. If the application that is writing to the log file continues to write to it, adTempus will continue to trigger your job each time it detects a change (it will check for changes roughly every 10 seconds).

See the File Trigger Overview topic for more information.

⭐Additional Options

Wait for exclusive access to the file

If this option is checked, adTempus will not trigger the job until it can get exclusive access to the file (i.e., no other application is reading or writing the file). Note: In some cases it may be more reliable to watch the last modification timestamp instead (see next section), because the writing program may repeatedly close and then reopen the file while writing.

This option is not available for triggers that use FTP or SFTP.

Wait until __ seconds after the file's last modification

If this option is checked, adTempus will wait until the file has not been modified within the specified number of seconds before it triggers the job. This is used to ensure that adTempus does not trigger for a file that is still being written by another application. In some cases this option may be more reliable than the **Wait for exclusive access** option.

Capture the file

If this option is checked, adTempus will capture the file that satisfied the trigger and save it in the job history.

This option is not available for triggers that use FTP or SFTP.

Selection Script

You may specify a script that is to be run to determine whether a file should cause the job to trigger. When you specify a script, adTempus evaluates files as follows:

1. It finds all files that match your file specifications.
2. It discards any files that do not meet other selection criteria you have specified.
3. For each remaining file, it calls the script you specify. Information about the file is passed to your script in Job Variables (see below) and in the Parameters.FoundFiles property object exposed to .NET scripts.
4. Your script must return True if adTempus should trigger for the file or False if it should not.

Job Variables

The File Trigger sets the following Job Variables, which can be used by the selection script or by other scripts or notification messages in the job.

Variable Name	Description
FileTrigger.FileName	The full name (including path) of the file.
FileTrigger.FileSize	The size of the file (in bytes).
FileTrigger.FileCreationTime	The date/time at which the file was created.
FileTrigger.FileLastAccessTime	The date/time at which the file was last accessed.

Variable Name	Description
FileTrigger.FileLastWriteTime	The date/time at which the file was last modified.
FileTrigger.FileAttributes	An integer indicating the file's attributes.
FileTrigger.FileAction	The action that caused the trigger: `FileCreated`, `FileDeleted`, or `FileModified`"

Related Concepts

File Trigger 228

Reference

File Specification Properties 235

File Specification Properties

Each File Specification defines a file (or set of files) that will trigger the job.

File Specification

Specify the file(s). You must specify the path and name, but you may use wildcards. For example:

- `c:\proddata\data1.log` looks for a specific file
- `c:\proddata\*.log` looks for any file with a ".log" extension in the `c:\proddata\` directory.

The **File Specification** may contain Job Variables, which will be expanded to resolve the file name.

The File Trigger cannot be used to monitor network drives mapped to a local drive letter. You must use the UNC path to the files rather than the drive letter. This is because drive letters mapped outside adTempus are not available to adTempus, and drive mappings you establish on the Resources page are not connected until after the job has been triggered. See the "Network Access for Jobs " on page 482 topic for more information.

Include subdirectories

If the **Include subdirectories** option is checked, adTempus will check subdirectories of the directory named in the **File Specification** for files that match the same name or pattern. For example if your **File Specification** is `c:\proddata\*.log`, adTempus will look for a file with the extension ".log" in the"`c:\proddata\` directory and all of its subdirectories.

Related Concepts

File Trigger 228

Reference

File Trigger Properties 230

Job Trigger

Job Trigger

This feature is new for adTempus version 4.

A **Job Trigger** is a trigger that causes a job to execute based on the success, failure, or active state of one or more other jobs. It is used to link jobs together in chains.

The Job Trigger combines some of the features of the Job Control Action (which can also be used to link jobs together) and the Job Condition. See the How To: Link Jobs Together topic for a discussion of the differences between the Job Trigger and the Job Control Action.

Related Concepts

Triggers 212

Related Topics

Job Condition 263
Job Control Action 284
Comparison of Triggers and Conditions 87
Triggers Page 138

Reference

Job Trigger Properties 236
Job Trigger Target Rule Properties 237
How To: Link Jobs Together 494

Job Trigger Properties

The **Job Trigger Properties** windows contains the settings for a Job Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

Job Trigger

Each Job Trigger has one or more target rules. Each rule specifies a job that the current job is dependent on, and the conditions that must be satisfied for that job (succeeded, failed, etc.). If you specify more than one target, you can configure the trigger to start the job when *any* of those rules is satisfied, or to wait until *all* are satisfied.

On this page you can Add, Edit, or Delete Target Rules.

Related Concepts

Job Trigger 236

Reference

Job Trigger Target Rule Properties 237
How To: Link Jobs Together 494

Job Trigger Target Rule Properties

The **Job Trigger Target Rule** window contains the settings for a single target (dependency job) for a Job Trigger.

Each rule defines a job that the current job is dependent on, and the rules for satisfying that dependency.

★Settings

Enable this rule

Uncheck this box to disable the rule. adTempus will ignore any rules that are not enabled.

Trigger based on job

Select the job to depend on. Only jobs that you have "List/Reference" permission for are listed. The job may be on the same computer as the job you are editing, or on a different computer. If the job is on a different computer, the other computer must be defined as a Linked Server in adTempus.

Rule

Specify the rule that the target instance must satisfy:

- **Job Started.** The trigger fires when the target job starts.
- **Job Ended.** The trigger fires when the target job ends.
- **Job succeeded.** The target instance of the job must have completed with a successful result.
- **Job failed.** The target instance of the job must have completed with a failure result.

Instance

Specify which instance of the target job adTempus should look at to see if the rule is met. The "Target Job" is the job that you selected in the **Depend on Job** section. The "Dependent Job" is the job that has the trigger (the job you are currently editing).

The following instance rules are available:

- **Most recent since previous execution of current job.** adTempus will look at the most recent instance of the target job that executed since the last execution of the dependent job.
- **Most recent since previous successful execution of current job.** adTempus will look at the most recent instance of the target job that executed since the last *successful* execution of the dependent job.
- **Most recent in cycle.** adTempus will look at the most recent instance of the target job that has the same Cycle ID as the dependent job.
- **Most recent instance since date/time specified in a Job Variable.** adTempus will retrieve the current value of the specified variable (which must be defined as a Date/Time variable) and then look at the most recent instance of the target job that executed since that date/time.
- **Most recent instance.**adTempus will look at the most recent instance of the target job, regardless of when that instance executed.

- ★ **Any since previous execution of current job.** adTempus will look at all instances that have completed since the last execution of the dependent job. If any of these instances matches the Rule, the condition will be satisfied.
- **Any instance since previous successful execution of current job.** adTempus will look at all instances that have completed since the last *successful* execution of the dependent job. If any of these instances matches the Rule, the condition will be satisfied.
- **Any instance in cycle.** adTempus will look at all instances of the target job that have the same Cycle ID as the dependent job.
- **Any instance since date/time specified in a Job Variable.** adTempus will retrieve the current value of the specified variable (which must be defined as a Date/Time variable) and then look at all instances of the target job that have executed since that date/time.

If you use one of the options based on the cycle, you must have Cycle IDs set up correctly as described in the Cycle ID topic.

See this this example for an illustration of the effects of the settings.

Ignore manual (on-demand) instances of job

When this option is checked, the condition will not look at any instances of the dependent job that were run on demand using the Run command when it is applying a "previous execution" rule.

Example

Example

Suppose you are editing job "Data Copy" and setting a condition that job "Data Extract" must have run successfully since the most recent instance of "Data Copy." During your normal schedule cycle, "Data Extract" runs at 7AM, and "Data Copy" runs at 3PM.

Today, however, "Data Copy" was run manually at 8AM.

At 3PM "Data Copy" is triggered and the condition is checked.

If the **Ignore manual (on-demand) instances** option is not checked, adTempus will see that the previous instance of "Data Copy" was at 8AM. Since "Data Extract" has not been run since then, the condition will not be met, and "Data Copy" will not run.

If the option is checked, however, adTempus will ignore the 8AM instance and see yesterday's 3PM instance as the most recent instance of "Data Copy." Since today's "Data Extract" job ran after that, the condition will be met, and "Data Copy" will run.

Related Concepts

Job Trigger 236

Reference

Job Trigger Properties	236
How To: Link Jobs Together	494

Process Trigger

Process Trigger

The Process Trigger allows you to trigger a job whenever external processes start, end, or exceed a specified memory limit.

For example, you may have a server process that occasionally fails. Using a Process Trigger, you can have adTempus monitor this process and run a job (to restart the process, send a notification message, etc.) when it fails.

Related Concepts

Triggers	212

Related Topics

Comparison of Triggers and Conditions	87
Triggers Page	138
Process Termination Task	196
Process Condition	272
Service Control Task	204

Reference

Process Trigger Properties	240

Process Trigger Properties

The **Process Trigger Properties** window contains the settings for a Process Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

Process Criteria

Processes

Specify the process names that adTempus should monitor. You must specify the executable name (e.g., "notepad.exe"). If there is more than one process on the computer with the same name, you may optionally include the full path of the version of the process that you want to monitor.

If you specify more than one process, adTempus will trigger the job whenever any of the processes you list satisfies the other criteria.

If you do not specify any processes, adTempus will trigger for every process on the system, which is probably not what you want.

Trigger Conditions

Select the conditions under which the job will be triggered. You must select at least one option.

Use a script to select processes

adTempus will run the script you specify. Information about the process will be passed to the script using the Parameters object and through Job Variables (see list below). Your script then evaluates the process information and returns **True** if adTempus should trigger the job, or **False** if it should not.

adTempus calls the script only for events that have satisfied all of the other criteria for the trigger.

Description/Notes

Enter any extended descriptive information or notes for this trigger. There is no limit on the length of the text.

Job Variables

The Process Trigger sets the following Job Variables, which can be used by a selection script or by other scripts or notification messages in the job.

Parameter Name	Description
ProcessTrigger.ProcessID	The Windows process ID for the process.
ProcessTrigger.ProcessName	The name of the process
ProcessTrigger.ProcessEvent	The action that caused the trigger: `TriggerProcessStarted`, `TriggerProcessEnded`, `TriggerProcessMemExceeds`.
ProcessTrigger.ProcessEventDescription	Description of the action that caused the trigger: "started", "ended", or "exceeded memory threshold"

Related Concepts

Process Trigger........240

Schedule Trigger

Schedule Trigger

A **Schedule Trigger** is a trigger that causes a job to be executed at specific dates and times, or at specific intervals (e.g., every 5 minutes).

When you create a Schedule Trigger, you specify one or more **Schedules**, which specify when the job should run. Options allow you to specify how adTempus should react to changes of the system clock. You may also specify what should happen if the job is scheduled to run on a holiday.

If a Schedule Trigger has more than one schedule, the job will be executed whenever any of the schedules is satisfied.

Multiple Schedules

A job can have more than one Schedule Trigger, and each Schedule Trigger can have any number of Schedules. The job will execute whenever any of the schedules on any of the Schedule Triggers is satisfied. However, if any Schedules or Schedule Triggers produce overlapping times, adTempus will still only run the job once. For example, even if two separate Schedules specify that the job should run at 12:30 PM on Monday, the job will only run once at that time.

You might want to have separate Schedule Triggers (instead of multiple Schedules on a single Schedule Trigger) if you want to use different time zone or holiday options for different schedules: all Schedules for a Schedule Trigger share the same options and holidays, but each separate Schedule Trigger for a job has its own time zone and holiday options.

Related Concepts

Triggers 212

Related Topics

Schedule Trigger Properties 243
Execution Schedule Properties 247
Comparison of Triggers and Conditions 87
Triggers Page 138

Schedule Trigger Properties

The **Schedule Trigger Properties** window contains the settings for a Schedule Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

Schedules

On the **Schedules** page you specify the Schedules that determine when the job should be run. You may add, edit, or remove schedules.

If you have specified more than one schedule, each schedule will trigger the job independently. For example, if one schedule includes all Mondays and another includes all Tuesdays, the job will be triggered on Mondays and Tuesdays.

If the schedules have overlapping times, this will *not* cause the job to be triggered twice at the same time. For example, if Schedule 1 and Schedule 2 both call for the job to be run at 12:30 PM on Monday, the job will only be run once at 12:30.

See the Schedule Trigger Overview topic for more information on using multiple schedules versus multiple Schedule Triggers.

Options

On the **Options** page you specify clock change and time zone options for this trigger.

System Clock Changes

These settings determine what adTempus will do if changes to the computer's clock affect scheduled execution times for the job.

Rerun if clock is set back

If this option is checked, the job will continue to run at its regular schedule even if the clock is set back so that executions are repeated. If the option is not checked, any "repeated" times will be ignored.

For example, your job is scheduled to run at 1:30 every morning. At 1:32 AM on 21 August 2002, just after the job has run, the system clock is set back to 1:29 AM when the clock is synchronized with a master clock. If the **Rerun...** option is checked, the job will be run when the clock reaches 1:30 again. If the option is not checked, the job will not be run again until 1:30 the next day.

Guidance: If your job is only meant to run once a day, you will want to leave this option unchecked (the default setting). If, however, your job is meant to run at regular intervals (e.g., to transfer data every 5 minutes), you will probably want to check the option so that the job continues to execute regularly even when the clock is set back by an hour for daylight saving time.

Run once if clock is set forward so that one or more scheduled executions are missed

If this option is checked, adTempus will run the job once if a time change causes a regular execution to be missed. If the option is not checked, executions may be missed as a result of the time change.

For example, your job is scheduled to run at 2:30 every morning. At 2:00 one morning the system clock is set forward to 3:00 am due to daylight saving time. If the **Run once...** option is checked, adTempus will run the job when the time change is detected to make up for the missed time. If the option is not checked, the execution will be missed.

adTempus will only execute the job once, regardless of the number of executions that are missed. For example, even if your job runs every minute and the clock has been set forward by a full hour, adTempus will run the job once, not 60 times.

Guidance: If your job only runs once a day (or less frequently) you will want to check this option (the default setting) to ensure that the job gets run. If, however, the job runs frequently (e.g., once an hour) you may want to uncheck this option, as missing one execution is not likely to be significant.

Time Zone

The time zone setting determines which time zone adTempus should evaluate this trigger in.

Local time zone on the computer where the job is triggered

The trigger will be evaluated in the time zone of the where the job is triggered. That is, if the job is scheduled to run at 11:30 AM, it will run at 11:30 AM local time on the computer triggering the job.

Note: If this job is in a Distributed Scheduling queue, this option causes adTempus to use the time zone of the Master computer for Slave and Load Balance queues (because the Master triggers the job for these configurations). For Mirror queues, the time zone of the Agent that runs the job will be used.

A specific time zone

The trigger will be evaluated in the specified time zone. For example, if the job is scheduled to run at 11:30 AM and you select the Pacific time zone, the job will run at 11:30 AM Pacific time. If the server running adTempus is located in the Eastern time zone, the job will execute at 2:30 PM local (Eastern) time.

The time zone of the Master with which the Agent is associated

This option is only available if you are configuring the job on a Distributed Scheduling Master computer. When this option is selected, the time zone of the Master will be used to evaluate the scheduled times, even for Mirrored jobs.

Holidays

On the **Holidays** page you specify how adTempus should treat this job if it is scheduled to run on a holiday.

Use the following set of holidays

If you want the job to behave differently on holidays, check this option and select the list of holidays that applies. You may optionally create or edit a holiday set if none of the existing sets meets your needs.

If you do not want the job to behave differently on holidays, uncheck this option.

When the job is scheduled to run on a holiday

Specify how adTempus should behave if a scheduled execution for the job falls on a holiday. The following options are available:

- **Run as scheduled.** The job is still run according to the schedule rules. Use this option if you using date selection rules to cause the job to run only on holidays or non-holidays. See the Date Rule Properties topic for more information.
- **Don't run the job.** The job is not run on any day that is defined as a holiday.
- **Run the job on the nearest business day.** The job is run on the nearest day that is not a weekend or holiday. This could be either before or after the originally scheduled date.
- **Run the job on the previous business day.** The job is run on the nearest prior day that is not a weekend or holiday.
- **Run the job on the next business day.** The job is run on the nearest subsequent day that is not a weekend or holiday.
- **Run the job on the nearest non-holiday.** The job is run on the nearest day that is not a holiday. This could be either before or after the originally scheduled date and could be a weekend.
- **Run the job on the previous non-holiday day.** The job is run on the nearest prior day that is not a holiday (but may be a weekend).
- **Run the job on the next non-holiday.** The job is run on the nearest subsequent day that is not a holiday (but may be a weekend)

Related Concepts

Schedule Trigger 242
Holiday Set 403

Related Topics

Execution Schedule Properties 247

Reference

Holiday Set Properties 404

Execution Schedule

An Execution Schedule defines the dates and times at which a job will run.

Schedules are attached to Jobs using Schedule Triggers. A trigger may have any number of schedules.

Reference

Execution Schedule Properties 247
Date Rule Properties 249

Execution Schedule Properties

The **Execution Schedule Properties** window allows you to view or edit the properties of a Schedule for a Schedule Trigger.

Property Pages

Date Selection

The **Date Selection** page defines the days on which the job will run. You may either use an existing Shared Schedule or specify dates for the job.

Enable this schedule

Check this option to make the schedule "active." If the Schedule is not enabled, it will be ignored by adTempus (jobs will not be triggered according to this schedule).

Optional Descriptive Name

You may optionally enter a descriptive name for this schedule. If none is entered, adTempus will use a name generated based on the settings.

Use this shared schedule

Check this option to use an existing Shared Schedule to determine the days on which the job will run. Select the shared schedule that you wish to use.

You still must specify the times the job will run, using the Time Selection page.

Date Selection

The date selection options are not available if you have chosen to use a shared schedule. There are two options for specifying the days that the job will run:

- **Trigger every __ days.** adTempus will trigger the job at the fixed interval you specify. The job will be triggered on the starting date you specify in the **Active Range** and every *n* days afterward (where *n* is the interval you have specified).
- **Trigger on specific days.** When you select this option you have complete flexibility to define as many specific dates or rules as you wish. Use the list to add, edit, or remove Date Rules that specify the days on which the job should run.

⭐Active Range

Specify the earliest date and time to which this schedule applies. Optionally, specify an ending date and time.

If you specify an ending date and time, the schedule will not be used (will not trigger jobs) after that date/time.

Sharing

The Sharing options are not available if you have chosen to use a shared schedule.

Once you have created a schedule for your job, you can share it to make it available for use in other jobs.

Make this schedule available for use by other jobs

Check this option to share the schedule.

> Once you have checked this option and saved your changes by clicking **OK**, your schedule becomes a Shared Schedule. When you return to this window you will find that the **Use this shared schedule** option is now checked and your schedule is selected in the shared schedule list. To modify the schedule you must edit it as a shared schedule.

Once a schedule has been shared, it cannot be "unshared" without deleting it.

Name for shared schedule

Provide a descriptive name for the schedule. The name is required, and must be unique across all existing shared schedules.

Time Selection

The **Time Selection**page defines the times at which the job will run.

Trigger every...

Select this option to execute the job at a fixed interval (e.g., every 5 minutes or every 2 hours). The first execution will occur at the starting time you specify.

Trigger at these times

Select this option to specify exact times at which the job should execute. You can list any number of times.

Randomize Start Time

Check this option to have adTempus randomize the start time. This can be used to ease the load on adTempus or the server. For example, if you have a large number of jobs that run every hour, you can configure them to start randomly within 5 minutes of the beginning of the hour.

When you check this option, adTempus will randomly choose a start time within the specified number of minutes *after* the scheduled start time. If you also check the **Allow job to start before scheduled time**, the job may be started within the specified number of minutes before the scheduled start time as well. That is, if the job is scheduled to run at 11:30 and you set a limit of 5 minutes and check the before option, the job will start somewhere between 11:25 and 11:35.

Related Concepts

Execution Schedule 246
Schedule Trigger 242

Related Topics

Schedule Trigger Properties 243

Reference

Date Rule Properties 249

Date Rule Properties

The **Date Rule Properties** window allows you to edit a date rule for an Execution Schedule within a Schedule Trigger. Options allow you to specify specific dates or rules that will determine the dates when a job will run.

Optional name for this selection

Optionally, specify a name for this selection to help identify it in lists.

Enabled

Check to enable this rule. If the box is unchecked, the rule is disabled and will be ignored by adTempus when scheduling jobs.

Months

Unless you are using the Easter option, you must select the month(s) in which the job should run.

Trigger On

Select the type of rule you want to use.

Selected days of month

Select this option to execute jobs on specific days of the month. The job will execute on the day(s) you select in each of the months you have selected in the **Months** list.

If days you select fall after the last day of a month, the job will execute on the last day. For example, if you have selected all of the months in the months list, and have specified "30" as the day on which to execute, the job will execute on the last day of February (28th or 29th), and on the 30th of the other months.

Selected days of week

The job will be executed on the days of the week you specify, in each of the months you have selected in the **Months** list.

The holiday settings for the Schedule Trigger determine which days, if any, are treated as holidays.

Use the **Recur every *x* weeks** option to run the job less frequently than every week. Specify 1 to run every week, 2 to run every other week, etc. This recurrence can be relative to the Active Range start date specified for the schedule, the beginning of the year, or the beginning of each selected month.

If you select more than one weekday, the job will trigger on each selected day. See Selecting Days of the Week below for more information.

Days specified using a floating rule

This option allows you to build complex rules to specify the date. This allows you to accommodate scenarios such as:

- The first business day of each month
- The third Tuesday of each month
- The day after the last business day of each month
- The Tuesday after the fourth Sunday of March
- The nearest business day to Christmas

To define your date:

1. Decide whether your floating date can be specified relative to the beginning of the month (as in the first four examples listed above) or must be specified relative to some specific or floating date (as in the Christmas example above).
   - If the date is relative to the beginning of the month, select the "relative to the beginning of the month" option.
   - If the date is relative to another date, select the "relative to a specific day..." option, then click the **Select...** button. This will display a new **Date Rule Properties** window, where you may use any of the available options to define your starting date.
2. Once you have decided on a starting point, decide which days of the week should serve as the basis for your rule, and select them.

   If you select more than one day, the job will trigger only on the first day that meets the other criteria. See Selecting Days of the Week below.
3. Then select the occurrence (1st, 2nd, 3rd, etc.) of that day within the month.

4. Finally, you may specify a number of days before or after the selected date (enter "0" to use the selected date itself).

Days specified relative to end of month

Use this option to run the job on the last day of each month, or a certain number of days before or after the last day.

Days specified relative to Easter

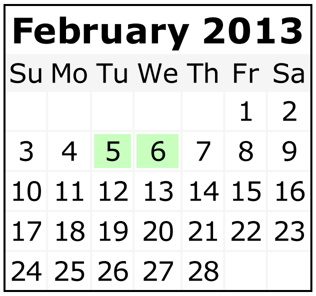
Use this option to run this job on Easter, or a specified number of days before or after Easter. adTempus calculates the date for Easter each year and runs the job accordingly.

Selecting Days of the Week

The "Selected days of week" and "Days specified using a floating rule" rules both allow you to select one or more days of the week when jobs should trigger. However, these two rules behave differently when you select more than one day.

Selected days of week

When the rule is "Selected days of the week," the job is triggered on *each* weekday that you select. For example, if you configure the rule to "Recur every 2 months relative to Beginning of month" and select "Tuesday" and "Wednesday," the job will trigger every other Tuesday *and* every other Wednesday. For February 2013, the job will trigger on February 5 and February 6:

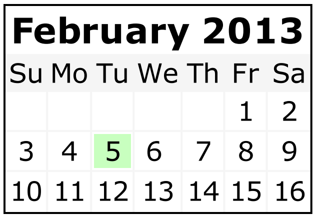


February 2013

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

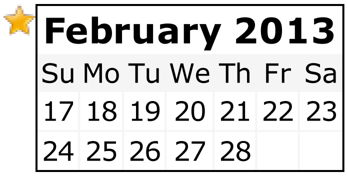
Days specified using a floating rule

When the rule is "Days specified using a floating rule," the job is triggered for only the first date that matches the criteria. For example, if you configure the rule to trigger on the first occurrence of the month and select "Tuesday" and "Wednesday," the job will trigger on the first day of the month that is either a Tuesday or a Wednesday. For February 2013, this is February 5:



February 2013

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16



February 2013						
Su	Mo	Tu	We	Th	Fr	Sa
17	18	19	20	21	22	23
24	25	26	27	28		

The job will *not* trigger on February 6.

Matching Weekdays, Business Days, Weekends, Holidays for Floating Day Rules

In adTempus 3 and earlier, there were pseudo-days that matched based on weekdays, business days, weekends, holidays, etc. These have been remove starting with adTempus 4 in favor of selecting the days that you want, in order to provide greater flexibility. For example, in earlier versions you could configure a schedule for the "3rd business day," which worked fine as long as your "business days" are Monday through Friday. If your business days are Sunday through Thursday, there was no way to make a rule for that. Starting with adTempus 4, you can. The following table shows the settings that should be used to match the old options.

If you use a setting like "Any Weekday" frequently, you should set up a Shared Schedule configured with the appropriate settings and then reference it from your schedules, so that you don't have to configure the options each time you create a new schedule.

Version 3 Setting	Version 4 Settings	Notes
Any Weekday	Check the boxes for the days that are "weekdays" (working days), e.g., Monday through Friday.	
Any Weekend Day	Check the boxes for the days that are weekend days (non-working days), e.g., Saturday and Sunday.	
Any Business Day	Check the boxes for the days that are "weekdays" (working days), e.g., Monday through Friday. Also check the **Exclude Holidays** box.	The Schedule Trigger must be configured with the appropriate holiday settings; see below.
Any Non-Business Day	Check the boxes for the days that are weekend days (non-working days), e.g., Saturday and Sunday. Also check the **Any Holiday** box.	The Schedule Trigger must be configured with the appropriate holiday settings; see below.
Any Holiday	Check the **Any Holiday** box and make sure all weekday boxes are unchecked.	The Schedule Trigger must be configured with the appropriate holiday settings; see below.
Any Non-Holiday	Check the **Any Non-Holiday** box and make sure all weekday boxes are unchecked.	The Schedule Trigger must be configured with the

Version 3 Setting	Version 4 Settings	Notes
		appropriate holiday settings; see below.

Holidays

In order for a schedule to match holidays or non-holidays, the Holidays page of the Schedule Trigger properties must have a Holiday Set selected to tell adTempus which holidays you want to apply. However, you should set the holiday rule on that page to **Run as scheduled** so that none of the special holiday handling rules are applied. This is because your Date Rule already has the holiday settings you want to use.

Related Concepts

Execution Schedule 246

Reference

Execution Schedule Properties 247

Startup Trigger

Startup Trigger

A Startup Trigger is a trigger that causes a job to be executed each time the adTempus service is started. This may be because the computer is being started, or because the adTempus service was restarted.

The adTempus service cannot determine whether it is being started as part of computer startup or because the adTempus service was stopped and restarted for some other reason. Therefore you may need to perform additional configuration to prevent duplicate processes, as discussed in the sections below.

Avoiding Duplicate Processes

If adTempus is stopped and restarted for a reason other than a computer restart, the program started by the job may still be running when adTempus restarts. When you use the Startup Trigger you should also use process conditions, the Process Execution task's **Skip if already running** option, or some other safeguard to make sure that your scheduled tasks do not get re-executed when they should not.

Example
Suppose you have a job with a Startup Trigger that is configured to start `OrderMonitor.exe`, which is intended to run as long as the computer is running. When

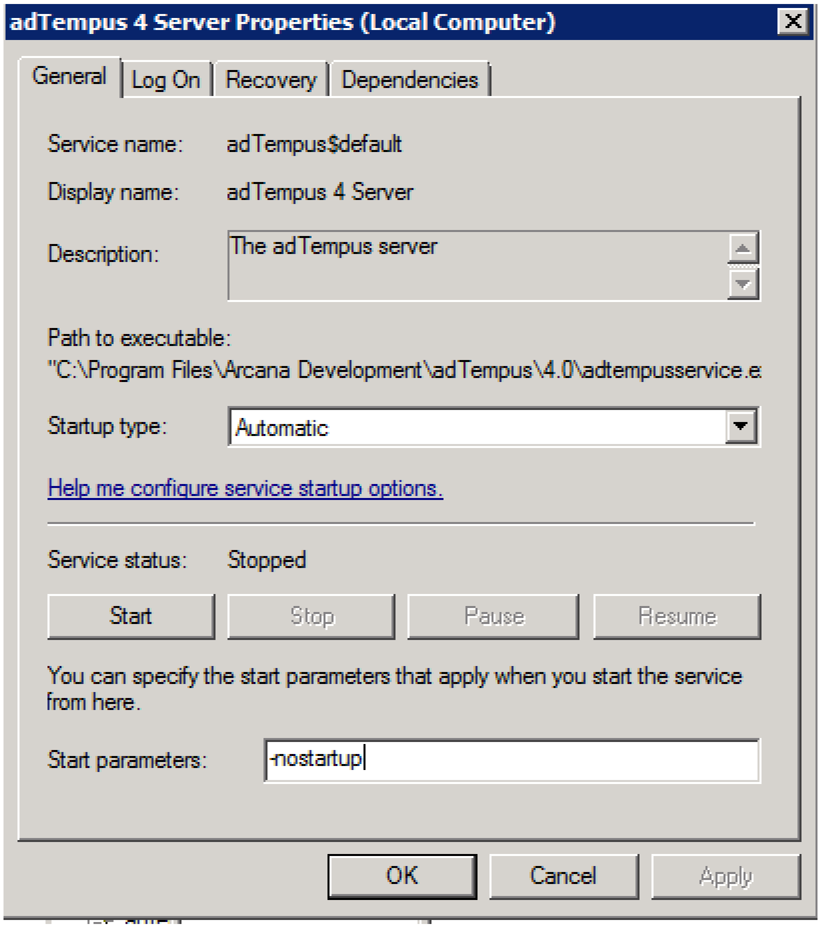
Windows starts, it starts the adTempus service, which starts `OrderMonitor.exe`.

Now suppose the adTempus service gets stopped and restarted for some reason (such as a software update). When the service starts again, it will again run the startup job, which will start `OrderMonitor.exe` again, unless you have configured one of the options discussed above.

Suppressing Startup Jobs

You can prevent adTempus from executing Startup Triggers by using the `-nostartup` **Start parameter** when starting the adTempus service from the Windows Services tool. (See the Service Startup Options topic for more information.)

When you specify this option, all Startup Triggers will be ignored.



adTempus 4 Server Properties (Local Computer)

Service name: adTempus$default

Display name: adTempus 4 Server

Description: The adTempus server

Path to executable:
"C:\Program Files\Arcana Development\adTempus\4.0\adtempusservice.e

Startup type: Automatic

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters: -nostartup

Related Concepts

Triggers 212

Related Topics

Comparison of Triggers and Conditions 87
Triggers Page 138

Reference

Startup Trigger Properties 255

Startup Trigger Properties

The **Startup Trigger Properties** window contains the settings for a Startup Trigger.

Property Pages

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

Startup Trigger

Wait ___ minutes before starting job

Specify how long (in minutes) adTempus should wait before starting the job. For example, if the program run by the job depends on other Windows system services (such as a database server), you may want to have adTempus wait a few minutes before starting the job, to ensure that the required services have been started.

You can also use Process Conditions to make sure that programs on which the job depends have been started.

Related Concepts

Startup Trigger 253

WMI Trigger

WMI Trigger

The WMI Trigger allows adTempus to receive Windows Management Instrumentation (WMI) events and trigger jobs based on them.

For information on WMI, refer to the Microsoft Platform SDK documentation.

For sample WMI queries, visit the Arcana Development web site.

Changes from Previous Versions

If you are upgrading from adTempus 3 or earlier and are using scripts to filter WMI events, some changes are required.

Filter scripts must now be .NET scripts (VB.NET or C#); WSH scripts (VBScript) are not supported. If your script is already a .NET script, the script class must be modified to derive from the WMITriggerScriptBase class. Previous versions of adTempus exposed the WMI event through a Job Variable. Starting with version 4 the event is passed in the Parameters.WMIEvent object instead, which exposes the event as a ManagementBaseObject . The event is no longer available through a Job Variable.

Related Concepts

Triggers 212

Related Topics

Comparison of Triggers and Conditions 87
Triggers Page 138

Reference

WMI Trigger Properties 256

WMI Trigger Properties

The **WMI Trigger Properties** window contains the settings for a WMI Trigger.

Property pages common to all Triggers

General

Name for this trigger (optional)

Optionally, specify a descriptive name for the trigger.

Enabled

Uncheck this box to disable the trigger. If the trigger is not enabled, adTempus will ignore it (it will not cause the job to trigger).

Minimum Interval

Specify the minimum interval (in seconds) for this trigger. Once the triggers conditions are satisfied and the job is triggered, adTempus will ignore this trigger for the specified interval.

For example, suppose you are using a File Trigger to trigger the job whenever a new file is added to a folder. If you set the interval to 60 seconds, then once adTempus triggers the job, it will ignore any new files that arrive in the next 60 seconds.

Description/Notes

Enter any extended descriptive information or notes for this trigger.

Job Variables

The Job Variables page allows you to define Job Variables that will be set for this job only when it is started by this trigger.

WMI Events

Specify the WMI query parameters.

Namespace

Specify the WMI namespace to use.

Query

Specify the WMI query to execute. This must be a valid WMI event query. For information on WMI event queries refer to the Microsoft Platform SDK documentation.

Each time the query returns an event, adTempus will trigger the job.

adTempus does not validate your WMI query settings at the time you save the trigger. To test your query, use the "wbemtest" system utility.

If Windows returns an error when adTempus submits your query, adTempus will log an error message to the message log for the job and hold the job until you edit it to correct the query.

Use a script to select events

For more advanced filtering, you can use a .NET script to filter the message. For each event returned by the query, adTempus will run your script. Your script has access to the Parameters.WMIEvent object, which exposes the event as a ManagementBaseObject.

Your script must return True if adTempus should trigger for the event or False if it should not.

If you are upgrading from adTempus 3 or earlier and are using scripts to filter WMI events, some changes are required: Show.

Filter scripts must now be .NET scripts (VB.NET or C#); WSH scripts (VBScript) are not supported. If your script is already a .NET script, the script class must be modified to derive from the WMITriggerScriptBase class. Previous versions of adTempus exposed the WMI event through a Job Variable. Starting with version 4 the event is passed in the Parameters.WMIEvent object instead, which exposes the event as a ManagementBaseObject . The event is no longer available through a Job Variable.

Conditions

Conditions

Conditions are used to place additional restrictions on whether a Job or Job Step should run. For example, you can specify that a job should run only if a particular file exists.

Conditions do not cause a job to start—only triggers do this. Conditions are evaluated *after* a job has been submitted for execution. See the Comparison of Triggers and Conditions topic for more information.

adTempus offers the following kinds of conditions:

- The Job Condition depends on the state of another job.
- The File Condition depends on the presence or absence of one or more files.
- The Process Condition depends on the state of an external process.
- The Job Variable Condition depends on the value of a Job Variable.
- The Script Condition allows you to use a script to write your own condition logic.

How Conditions Work

Conditions can be attached to Jobs and/or Job Steps.

When a job is submitted for execution (by a Trigger, a Response, through on-demand execution, etc.) adTempus checks to see if there are conditions for the job. If so, it evaluates each condition to see if the rules of that condition have been met (the condition is "satisfied").

If all conditions are met, execution continues. If one ore more conditions are not met, the settings on the Conditions page of the Job Properties determine whether adTempus will wait until the conditions are met, how long it will wait, and what will happen if the conditions are never met. If the conditions are not met, you can choose to have adTempus issue a warning and execute the job anyway, skip the job, or fail the job.

If a Job Step has one or more conditions, they are evaluated in the same way. Each step's conditions are evaluated only after the previous step has completed execution.

Responding to Unmet Conditions

You can use Responses to take action if one or more conditions are not met for a Job or Job Step. To do so, add the "One or more conditions failed" Event to the Response.

Adding and Editing Conditions

Conditions for a Job are managed on the Conditions page of the Job Properties window. Conditions for a Job Step are managed on the Conditions page of the properties window for the Step.

Ignoring Conditions

In some case you may want to temporarily ignore conditions without deleting them. There are several ways to do this:

- Edit the condition and uncheck the **Enabled** box on the **General** page of the condition properties. This will cause adTempus to always ignore the condition.
- If you are running the job on demand using the Run command, you can choose to ignore job-level and/or step-level conditions.
- When you configure a Job Control Action to run a job, you can configure it to ignore job-level and/or step-level conditions

Viewing the Status of Conditions

While a job is waiting for conditions to be met, you can view the status of each condition to see which ones are met, and what adTempus is waiting for. To do this, select the waiting instance in the History panel or the Job Monitor and display the properties for the instance. The properties window has a Condition Status page, which shows the status of each condition.

Related Concepts

Key Concepts 81

Related Topics

Comparison of Triggers and Conditions 87

File Condition

File Condition

A File Condition is a condition that depends on the presence or absence of a specified file or group of files.

For example, you have a job processes data files that are transferred to your server by FTP. You want the job run each hour, but only if new files have been transferred. You attach a File Condition to the job so that it will only run if files exist.

A File Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for files. To start a job based on files, use a File Trigger instead.

Related Concepts

Conditions 258

Related Topics

File Trigger 228
File Operation Tasks 161
Comparison of Triggers and Conditions 87

Reference

File Condition Properties 260

File Condition Properties

The **File Condition Properties** window contains the settings for a File Condition.

A File Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for files. To start a job based on files, use a File Trigger instead.

Property Pages

Property pages common to all Conditions

General

Name for this condition (optional)

Optionally, enter a name for this condition.

Enabled

Check this box to enable the condition, or uncheck to disable it. A disabled condition will be ignored by adTempus.

Description/Notes

Optionally, enter an extended description or notes about this condition.

Condition Wait

The options on this page determine how adTempus waits for this condition.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Timed Out."

Condition Polling

This option determines how often this condition should be evaluated. In most cases you will use the default polling interval (about 30 seconds). You may wish to use a longer polling interval to reduce the load on adTempus in cases where checking a condition is resource-intensive (such as a Script Condition that does a large amount of processing).

When you set a polling interval for one or more conditions on a job or step, adTempus applies the longest polling interval to all conditions.

The server-wide default interval can be set through the Advanced Server Options window.

Once satisfied, do not re-evaluate when evaluating other conditions for this instance

This option determines how this condition behaves once it is met if the job or step also has other conditions that have not been met. adTempus will keep checking each condition until it is met or times out.

If this option is checked, adTempus will not check the condition again once it has been met. If the option is not checked, adTempus *will* keep checking the condition each time it checks the other conditions.

Example

Example

Suppose your job has two conditions:

1. File `c:\input\*.zip` must exist.
2. Job "Data Process" must not be running.

Both are configured to wait up to 10 minutes for the condition to be met, and the job is configured to wait until all conditions are met.

The first time it checks conditions, adTempus finds file `c:\input\file1.zip`, so the condition 1 is now met. Job "Data Process" is running, so condition 2 is not met. adTempus therefore must wait for condition 2 to be met, so 30 seconds later it goes to check the conditions again. During those 30 seconds, job "Data Process" has finished running (so condition 2 has now been met), but the file `c:\input\file1.zip` has been deleted.

If condition 1 has the **Once satisfied...** option checked, adTempus will not re-evaluate the condition, so it considers the condition to be satisfied even though the file no longer exists. The job is therefore triggered.

If condition 1 does *not* have the **Once satisfied...** option checked, adTempus will re-evaluate the condition and find that the file no longer exists. Condition 1 therefore is no longer satisfied, so adTempus will continue waiting until it is satisfied again, or until the wait time elapses.

File Condition

File Specification

Specify the files that the job is dependent on. You must specify the path and name, but you may use wildcards. For example:

- "c:\proddata\data1.log" looks for a specific file
- "c:\proddata\*.log" looks for any file with a ".log" extension in the "c:\proddata\" directory.

The **File Specification** may contain Job Variables, which will be expanded to resolve the file name.

Criterion

Specify what the state of the file must be to satisfy the condition:

- **File must exist**. The file you specify must exist. If your **File Specification** includes wildcards, the condition is satisfied as soon as one file that matches the pattern is found.
- **File must not exist.** The file you specify must not exist. If your **File Specification** includes wildcards, the condition is satisfied only if no files that match the pattern are found.

Include subdirectories

If the **File Specification** contains wildcards and the **Include subdirectories** option is checked, adTempus will check subdirectories of the directory named in the **File Specification**. For example if your **File Specification** is `c:\proddata\*.log`, adTempus will look for a file with the extension ".log" in the "c:\proddata\" directory and all of its subdirectories.

Only consider files modified since the last execution of the job

If this option is checked, adTempus will only look at files modified (or created) since the last time this job was executed.

Wait for exclusive access to the file

If this option is checked the condition is not satisfied until adTempus can get exclusive access to the file (that is, the file is not in use by any other processes).

For example, your job is dependent on a data file transferred to the server using FTP. The transfer process can take several minutes to complete. The file will appear in the directory as soon as the transfer begins, which would cause the condition to be satisfied. However, you don't want the job to run until the transfer is complete. If you check the **Wait for exclusive access to the file** option, adTempus will wait (as long as you have also checked the **Wait up to __ seconds for the condition to be met** option) for exclusive access to the file. As long as the file is still being written to, adTempus will not be able to gain exclusive access and so will keep waiting. Once the transfer is complete, adTempus will be able to gain exclusive access and will allow the job to continue.

Related Concepts

File Condition 260

Job Condition

Job Condition

A **Job Condition** is a condition that prevents a job or step from executing unless another job is in a specified state.

For example, you have a job that reprocesses the data cubes for your data warehouse. This job cannot run until the three jobs that extract the data for the data warehouse have run. You therefore add a Job Condition to your cube reprocess job, so that it waits on the three extract jobs.

A Job Condition will not cause a job to start executing: it will only cause a job to pause execution to wait for other jobs. Some other mechanism (a Trigger or a Response on another job) must start the job. If you want one job to cause another to start when it finishes, you probably want to use a Job Trigger or a Job Control Action instead of a Job Condition. See the Comparison of Triggers and Conditions topic for more information about the differences.

Related Concepts

Conditions 258

Related Topics

Job Control Action 284
Job Trigger 236
Job Condition Instance Example 268
Comparison of Triggers and Conditions 87

Reference

Job Condition Properties 264

Job Condition Properties

The **Job Condition Properties** window is shown when you create or edit a Job Condition for a Job or Job Step.

Property Pages

Property pages common to all Conditions

General

Name for this condition (optional)

Optionally, enter a name for this condition.

Enabled

Check this box to enable the condition, or uncheck to disable it. A disabled condition will be ignored by adTempus.

Description/Notes

Optionally, enter an extended description or notes about this condition.

Condition Wait

The options on this page determine how adTempus waits for this condition.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."

- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Timed Out."

Condition Polling

This option determines how often this condition should be evaluated. In most cases you will use the default polling interval (about 30 seconds). You may wish to use a longer polling interval to reduce the load on adTempus in cases where checking a condition is resource-intensive (such as a Script Condition that does a large amount of processing).

When you set a polling interval for one or more conditions on a job or step, adTempus applies the longest polling interval to all conditions.

The server-wide default interval can be set through the Advanced Server Options window.

Once satisfied, do not re-evaluate when evaluating other conditions for this instance

This option determines how this condition behaves once it is met if the job or step also has other conditions that have not been met. adTempus will keep checking each condition until it is met or times out.

If this option is checked, adTempus will not check the condition again once it has been met. If the option is not checked, adTempus *will* keep checking the condition each time it checks the other conditions.

Example

Example

Suppose your job has two conditions:

1. File `c:\input\*.zip` must exist.
2. Job "Data Process" must not be running.

Both are configured to wait up to 10 minutes for the condition to be met, and the job is configured to wait until all conditions are met.

The first time it checks conditions, adTempus finds file `c:\input\file1.zip`, so the condition 1 is now met. Job "Data Process" is running, so condition 2 is not met. adTempus therefore must wait for condition 2 to be met, so 30 seconds later it goes to check the conditions again. During those 30 seconds, job "Data Process" has finished running (so condition 2 has now been met), but the file `c:\input\file1.zip` has been deleted.

⭐If condition 1 has the **Once satisfied...** option checked, adTempus will not re-evaluate the condition, so it considers the condition to be satisfied even though the file no longer exists. The job is therefore triggered.

If condition 1 does *not* have the **Once satisfied...** option checked, adTempus will re-evaluate the condition and find that the file no longer exists. Condition 1 therefore is no longer satisfied, so adTempus will continue waiting until it is satisfied again, or until the wait time elapses.

Job Condition

These options determine which job the condition will wait for, and which instance of the job.

Depend on Job

Select the job to depend on. Only jobs that you have "List/Reference" permission for are listed. The job may be on the same computer as the job you are editing, or on a different computer. If the job is on a different computer, the other computer must be defined as a Linked Server in adTempus.

Rule

Specify the rule that the target instance must satisfy:

- **Job must be running.** The target job must currently be running.
- **Job must not be running.** The target job must not currently be running.
- **Job must have succeeded.** The target instance of the job must have completed with a successful result.
- **Job must have failed.** The target instance of the job must have completed with a failure result.
- **Job must have completed.** The target instance of the job must have run, but the result is not important.
- **Job has not run.** The target job must not have run. This rule can be used to take action if a job does not run on time.
- **Job has not run successfully.** The target job must not have run successfully. This rule can be used to take action if a job does not run on time.

Instance

Specify which instance of the target job adTempus should look at to see if the condition is met. The "Target Job" is the job that you selected in the **Depend on Job** section. The "Dependent Job" is the job that has the condition (the job you are currently editing).

The following instance rules are available:

- **Most recent since previous execution of current job.** adTempus will look at the most recent instance of the target job that executed since the last execution of the dependent job.

- **Most recent since previous successful execution of current job.** adTempus will look at the most recent instance of the target job that executed since the last *successful* execution of the dependent job.
- **Most recent in cycle.** adTempus will look at the most recent instance of the target job that has the same Cycle ID as the dependent job.
- **Most recent instance since date/time specified in a Job Variable.** adTempus will retrieve the current value of the specified variable (which must be defined as a Date/Time variable) and then look at the most recent instance of the target job that executed since that date/time.
- **Most recent instance.**adTempus will look at the most recent instance of the target job, regardless of when that instance executed.
- **Any since previous execution of current job.** adTempus will look at all instances that have completed since the last execution of the dependent job. If any of these instances matches the Rule, the condition will be satisfied.
- **Any instance since previous successful execution of current job.** adTempus will look at all instances that have completed since the last *successful* execution of the dependent job. If any of these instances matches the Rule, the condition will be satisfied.
- **Any instance in cycle.** adTempus will look at all instances of the target job that have the same Cycle ID as the dependent job.
- **Any instance since date/time specified in a Job Variable.** adTempus will retrieve the current value of the specified variable (which must be defined as a Date/Time variable) and then look at all instances of the target job that have executed since that date/time.

If you use one of the options based on the cycle, you must have Cycle IDs set up correctly as described in the Cycle ID topic.

See this this example for an illustration of the effects of the settings.

Ignore manual (on-demand) instances of job

When this option is checked, the condition will not look at any instances of the dependent job that were run on demand using the Run command when it is applying a "previous execution" rule.

Example

Example

Suppose you are editing job "Data Copy" and setting a condition that job "Data Extract" must have run successfully since the most recent instance of "Data Copy." During your normal schedule cycle, "Data Extract" runs at 7AM, and "Data Copy" runs at 3PM.

Today, however, "Data Copy" was run manually at 8AM.

At 3PM "Data Copy" is triggered and the condition is checked.

If the **Ignore manual (on-demand) instances** option is not checked, adTempus will see that the previous instance of "Data Copy" was at 8AM. Since "Data Extract" has not been run since then, the condition will not be met, and "Data Copy" will not run.

If the option is checked, however, adTempus will ignore the 8AM instance and see yesterday's 3PM instance as the most recent instance of "Data Copy." Since today's "Data Extract" job ran after that, the condition will be met, and "Data Copy" will run.

Related Concepts

Job Condition 263

Related Topics

Job Condition Instance Example 268

Job Condition Instance Example

This example illustrates the behavior of the Job Condition based on the four available **Instance** options.

Suppose you are making job "Data Copy" dependent on job "Data Extract," with the Rule set to "Job must have succeeded" (that is, "Data Copy" cannot run until "Data Extract" has run successfully).

At 12:00 PM instance 312 of job "Data Copy" finishes.

At 12:15 PM instance 403 of job "Data Extract" finishes successfully.

At 12:30 PM instance 404 of job "Data Extract" fails.

At 12:45 PM instance 313 of job "Data Copy" starts.

- If the **Instance** is set to "The most recent instance," the condition is not met, because the most recent instance (404) failed.
- If the **Instance** is set to "The most recent instance since the previous execution of job," the condition is not met, because the most recent instance (404) failed.
- If the **Instance** is set to "Any instance since the previous execution of job," the condition is satisfied, because instance 403 succeeded.

At 1:00 PM instance 405 of job "Data Extract" finishes successfully.

At 1:15 PM instance 314 of job "Data Copy" starts. Regardless of which instance you have selected, the condition is met.

At 1:30 PM instance 315 of job "Data Copy" starts (note that "Data Extract" has not run again in the interim).

- If the **Instance** is set to "The most recent instance," the condition is met, because the most recent instance (405) succeeded. However, "Data Copy" has not been run since the last "Data Extract," so you probably would not want the "Data Copy" to run again yet.
- If the **Instance** is set to "The most recent instance since the previous execution of job," the condition is not met, because "Data Extract" has not been run since the last time "Data Copy" was run.
- If the **Instance** is set to "Any instance since the previous execution of job," the condition is not met, because "Data Extract" has not been run since the last time "Data Copy" was run.

Related Concepts

Job Condition 263

Reference

Job Condition Properties 264

Job Variable Condition

Job Variable Condition

A Job Variable Condition allows you to make a Job or Job Step conditional on the value of a Job Variable.

You can use it in conjunction with the Job Variable Task and Job Variable Action, which allow jobs to update Job Variables.

Related Concepts

Conditions 258
Job Variables 88

Related Topics

Comparison of Triggers and Conditions 87
How To: Set and Retrieve Variables in Scripts 500
Job Variable Update Action 289
Job Variable Update Task 178
Predefined Variables 485
Text Edit Window 441

Reference

Job Variable Condition Properties 270

Job Variable Properties 412

Job Variable Condition Properties

The **Job Variable Condition Properties** window is shown when you create or edit a Job Variable Condition for a Job or Job Step.

Property Pages

Property pages common to all Conditions

General

Name for this condition (optional)

Optionally, enter a name for this condition.

Enabled

Check this box to enable the condition, or uncheck to disable it. A disabled condition will be ignored by adTempus.

Description/Notes

Optionally, enter an extended description or notes about this condition.

Condition Wait

The options on this page determine how adTempus waits for this condition.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Timed Out."

Condition Polling

This option determines how often this condition should be evaluated. In most cases you will use the default polling interval (about 30 seconds). You may wish to use a longer polling interval to reduce the load on adTempus in cases where checking a condition is resource-intensive (such as a Script Condition that does a large amount of processing).

When you set a polling interval for one or more conditions on a job or step, adTempus applies the longest polling interval to all conditions.

The server-wide default interval can be set through the Advanced Server Options window.

Once satisfied, do not re-evaluate when evaluating other conditions for this instance

This option determines how this condition behaves once it is met if the job or step also has other conditions that have not been met. adTempus will keep checking each condition until it is met or times out.

If this option is checked, adTempus will not check the condition again once it has been met. If the option is not checked, adTempus *will* keep checking the condition each time it checks the other conditions.

Example

Example

Suppose your job has two conditions:

1. File `c:\input\*.zip` must exist.
2. Job "Data Process" must not be running.

Both are configured to wait up to 10 minutes for the condition to be met, and the job is configured to wait until all conditions are met.

The first time it checks conditions, adTempus finds file `c:\input\file1.zip`, so the condition 1 is now met. Job "Data Process" is running, so condition 2 is not met. adTempus therefore must wait for condition 2 to be met, so 30 seconds later it goes to check the conditions again. During those 30 seconds, job "Data Process" has finished running (so condition 2 has now been met), but the file `c:\input\file1.zip` has been deleted.

If condition 1 has the **Once satisfied...** option checked, adTempus will not re-evaluate the condition, so it considers the condition to be satisfied even though the file no longer exists. The job is therefore triggered.

If condition 1 does *not* have the **Once satisfied...** option checked, adTempus will re-evaluate the condition and find that the file no longer exists. Condition 1 therefore is no longer satisfied, so adTempus will continue waiting until it is satisfied again, or until the wait time elapses.

Condition

Variable to test

Select the variable this condition should check. This must be a variable that has already been defined in adTempus.

Comparison rule

Specify the comparison rule to apply. The comparison will be applied based on the type of the target variable. For example, if the variable is defined as a Date/Time variable, adTempus will treat the **Compare to** value as a date/time and compare the values as date/time values. If the variable type is unspecified, or the **Compare to** value cannot be converted to the correct type, the values will be compared as text.

Compare to

Enter the value to compare the target variable to. You can insert references to other variables here. If the target variable is configured as Date, Time, or Date/Time, use the following formats:

- Date: Use `YYYY-MM-dd` format ("2013-05-01")
- Time: Use `HH:mm:ss` format, using 24-hour time ("13:45" for 1:45 PM)
- Date/Time: Use `YYYY-MM-dd HH:mm:ss` ("2013-05-01 13:45:23")

Text comparisons are not case-sensitive.

Use Regular Expression

If the **Variable to test** is a string variable, check this option to treat the **Compare To** value as a Regular Expression to match against the variable value.

Related Concepts

Job Variable Condition 269

Process Condition

Process Condition

A Process Condition is a condition that depends on the state of a Windows process (program). Unlike a Job Condition the Process Condition allows you to wait on a process that is not under the control of adTempus.

For example, you have a job that requires exclusive access to a data file. This file is also used by a process that is often run by users outside of adTempus. By adding a Process Condition that targets this external process you can prevent adTempus from running the job when the data file is in use.

If you want to prevent a Program Execution Task from running if the target process is already running, the Skip this step if the process is already running option provides a one-step alternative to the Process Condition.

Related Concepts

Conditions 258

Related Topics

Process Termination Task 196
Process Trigger 240
Service Control Task 204
Comparison of Triggers and Conditions 87

Reference

Process Condition Properties 273

Process Condition Properties

The **Process Condition Properties** window contains the settings for a Process Condition.

Property Pages

Property pages common to all Conditions

General

Name for this condition (optional)

Optionally, enter a name for this condition.

Enabled

Check this box to enable the condition, or uncheck to disable it. A disabled condition will be ignored by adTempus.

Description/Notes

Optionally, enter an extended description or notes about this condition.

Condition Wait

The options on this page determine how adTempus waits for this condition.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that

time limit, job execution will continue immediately (as depending on other conditions). If the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Timed Out."

Condition Polling

This option determines how often this condition should be evaluated. In most cases you will use the default polling interval (about 30 seconds). You may wish to use a longer polling interval to reduce the load on adTempus in cases where checking a condition is resource-intensive (such as a Script Condition that does a large amount of processing).

When you set a polling interval for one or more conditions on a job or step, adTempus applies the longest polling interval to all conditions.

The server-wide default interval can be set through the Advanced Server Options window.

Once satisfied, do not re-evaluate when evaluating other conditions for this instance

This option determines how this condition behaves once it is met if the job or step also has other conditions that have not been met. adTempus will keep checking each condition until it is met or times out.

If this option is checked, adTempus will not check the condition again once it has been met. If the option is not checked, adTempus *will* keep checking the condition each time it checks the other conditions.

Example

Example

Suppose your job has two conditions:

1. File `c:\input\*.zip` must exist.
2. Job "Data Process" must not be running.

Both are configured to wait up to 10 minutes for the condition to be met, and the job is configured to wait until all conditions are met.

The first time it checks conditions, adTempus finds file `c:\input\file1.zip`, so the condition 1 is now met. Job "Data Process" is running, so condition 2 is not met. adTempus therefore must wait for condition 2 to be met, so 30 seconds later it goes to check the conditions again. During those 30 seconds, job "Data Process" has finished running (so condition 2 has now been met), but the file `c:\input\file1.zip` has been deleted.

If condition 1 has the **Once satisfied...** option checked, adTempus will not re-evaluate the condition, so it considers the condition to be satisfied even though the file no longer exists. The job is therefore triggered.

If condition 1 does *not* have the **Once satisfied...** option checked, adTempus will re-evaluate the condition and find that the file no longer exists. Condition 1 therefore is no longer satisfied, so adTempus will continue waiting until it is satisfied again, or until the wait time elapses.

Process Condition

Process Name

Specify the process names that adTempus should depend on. You must specify the executable name (e.g., "notepad.exe"). If there is more than one process on the computer with the same name, you may optionally include the full path of the version of the process that you want to monitor.

Note that this must be an executable (.exe). The process condition cannot target batch files (.bat, .cmd), scripts (.vbs, .js), etc.

Criterion

Specify what the state of the process must be to satisfy the condition:

- **Process must be running**. The process you specify must be running.
- **Process must not be running.** The process you specify must not be running.

Related Concepts

Process Condition 272

Script Condition

Script Condition

A Script Condition is a condition that executes a script. The result of the script determines whether the condition is satisfied.

Use a Script Condition to provide your own condition checks when none of the condition types provided with adTempus meets your needs.

For example, you want a job to execute only if a specified file contains certain text. The adTempus File Condition can detect the presence of the file but does not read its contents. Using a Script Condition you can create a script (using a scripting language like VB.NET) that looks for the file and then checks it for the required text.

Script Implementation Guidelines

Your condition script performs whatever processing is necessary to determine whether the condition is met, and then returns either a True (condition met) or False (condition not met) result to adTempus.

Waiting/Polling

All adTempus conditions, including the Script Condition, work on the principle of polling: each condition is checked repeatedly at an interval until it is met or the wait time limit expires. Therefore if your script determines that the condition is not met, it should return False to adTempus. adTempus will run your script again when the polling interval elapses, at which point the script should repeat the check.

Your script *should not* do any waiting itself. For example, suppose your script is making a call to a web service to see if remote processing has completed. Your script should not wait/loop if it finds that remote processing is not complete. Instead it should return immediately and allow adTempus to call it again when the polling interval elapses.

State Persistence

If your script needs to keep track of information between calls, you can do this using Job Variables. Any variables your script sets will be available the next time the script is called.

For example, the following VB.NET code fragment shows how to use a Job Variable named "CallCount" to track how many times the script has been called for the current job instance:

```
Dim callCount As Integer = 1
Dim variable As IJobVariable
variable=adTempus.JobVariables.GetVariable("CallCount")

If variable Is Nothing Then
       'script hasn't been called before
       'create the variable and set it to 1
       variable=adTempus.JobVariables.Add("CallCount")
       variable.IntegerValue=1
Else
       'get the previous value and increment it
       callCount=variable.IntegerValue.GetValueOrDefault() + 1
       variable.IntegerValue=callCount
End If
```

Reporting Additional Status Information

The Conditions page of the Instance Details window shows the status of each condition for the job or step ("Satisfied," "Waiting," etc.) and can also display an additional message about the status. Your script can specify the message to be displayed here by calling the SetConditionStatus method that is exposed by your script's base class:

```
SetConditionStatus("Additional condition status information")
```

For example, if your script determines that the condition is not yet met, it can set a status message indicating why (e.g., "Remote processing has not completed.")

The status is only updated when your script returns control to adTempus (the status is not updated immediately when you call the method). Therefore your script cannot report more than one status messages during a single execution of the script.

Related Concepts

Conditions 258

Related Topics

Comparison of Triggers and Conditions 87

Reference

Script Condition Properties 277

Script Condition Properties

The **Script Condition Properties** window contains the settings for a Script Condition.

Property Pages

Property pages common to all Conditions

General

Name for this condition (optional)

Optionally, enter a name for this condition.

Enabled

Check this box to enable the condition, or uncheck to disable it. A disabled condition will be ignored by adTempus.

Description/Notes

Optionally, enter an extended description or notes about this condition.

Condition Wait

The options on this page determine how adTempus waits for this condition.

Condition Wait

This option determines whether adTempus will wait for the condition to be met:

- **Do not wait for condition to be met.** If the condition is not satisfied the first time adTempus checks it for a given instance, adTempus treats the condition as "Failed."
- **Wait until the condition is met.** If the condition is not satisfied, adTempus will wait until it is satisfied. The job or step will show a status of "Waiting for Condition."
- **Wait up to __ minutes for the condition to be met.** adTempus will wait up to the specified time limit for the condition to be satisfied. If the condition is satisfied within that time limit, job execution will continue immediately (as depending on other conditions). If

the condition is not satisfied at the end of the time limit, adTempus treats the condition as "Timed Out."

Condition Polling

This option determines how often this condition should be evaluated. In most cases you will use the default polling interval (about 30 seconds). You may wish to use a longer polling interval to reduce the load on adTempus in cases where checking a condition is resource-intensive (such as a Script Condition that does a large amount of processing).

When you set a polling interval for one or more conditions on a job or step, adTempus applies the longest polling interval to all conditions.

The server-wide default interval can be set through the Advanced Server Options window.

Once satisfied, do not re-evaluate when evaluating other conditions for this instance

This option determines how this condition behaves once it is met if the job or step also has other conditions that have not been met. adTempus will keep checking each condition until it is met or times out.

If this option is checked, adTempus will not check the condition again once it has been met. If the option is not checked, adTempus *will* keep checking the condition each time it checks the other conditions.

Example

Example

Suppose your job has two conditions:

1. File `c:\input\*.zip` must exist.
2. Job "Data Process" must not be running.

Both are configured to wait up to 10 minutes for the condition to be met, and the job is configured to wait until all conditions are met.

The first time it checks conditions, adTempus finds file `c:\input\file1.zip`, so the condition 1 is now met. Job "Data Process" is running, so condition 2 is not met. adTempus therefore must wait for condition 2 to be met, so 30 seconds later it goes to check the conditions again. During those 30 seconds, job "Data Process" has finished running (so condition 2 has now been met), but the file `c:\input\file1.zip` has been deleted.

If condition 1 has the **Once satisfied...** option checked, adTempus will not re-evaluate the condition, so it considers the condition to be satisfied even though the file no longer exists. The job is therefore triggered.

If condition 1 does *not* have the **Once satisfied...** option checked, adTempus will re-evaluate the condition and find that the file no longer exists. Condition 1 therefore is no longer satisfied, so adTempus will continue waiting until it is satisfied again, or until the wait time elapses.

Script Condition

Script to evaluate

Select or create the script that adTempus should evaluate. See the Script Overview topic for information on working with scripts.

See the Script Condition overview for guidelines on implementing a condition script.

Your script must return a value of either **True** (condition is satisfied) or **False**(condition is not satisfied). Any other value will cause the condition to fail.

Related Concepts

Script Condition 275

Responses, Events, and Actions

Responses

Responses allow you to customize the flow of execution through an adTempus job, or to link jobs together.

While it is executing a job, adTempus fires "events" at certain key points. For example, an event is fired at the beginning of the job; another is fired at the end of the job. If the job completes successfully a "success" event is fired. If the job fails, a "failure" event is fired.

If you want adTempus to take action based on one of these events, you associate a Response with that event. The Response defines the actions that adTempus should take in response to the event. See an example.

Each response can be triggered by any number of events, and can execute any number of actions.

The events that are available vary based on whether you are creating responses for a job or a step, and based on the kind of task a step executes. For a complete list of events for a job or task, see the help topic for the Responses property page for the job or task.

Notes on using Responses

Responses are evaluated in the order that they appear in the Responses list for the job or step. Within a response, actions are executed in the order that they appear in the Actions list.

Actions are executed sequentially, one at a time. That is, each action must finish before the next action starts. Each Response's actions must complete before the next Response is evaluated. And all responses must be evaluated before job execution proceeds to the next step.

Related Topics

Actions 281
Events 280

Reference

Response Properties 280

Response Properties

A Response consists of events and actions. A Response is triggered when any of the events in its Events list occurs; when a Response is triggered all of its Actions are executed in the order that they are listed.

Events

Add, modify, or remove events that trigger this Response. See the Response Event topic for more information about specifying events. The help topic for the Responses page of the job or task for which you are adding responses describes the events that are available for the job or task.

You can specify any number of events for the Response. The Response will execute each time any of its events occurs. That is, if you have selected the events "Job Begin" and "Job End," the Response will execute once when the job begins and again when the job ends.

Actions

Add, modify, remove, or reorder the Actions that will be executed when this response is triggered. See the Actions topic for information on the available actions.

The actions will be executed sequentially in the order they appear in the list. Each Action must complete before the next Action executes.

Related Concepts

Responses, Events, and Actions 279

Related Topics

Actions 281
Events 280

Events

While it is executing a job, adTempus fires "events" at certain key points. For example, an event is fired at the beginning of the job; another is fired at the end of the job. If the job completes successfully a "success" event is fired. If the job fails, a "failure" event is fired.

The events that are available vary based on whether you are creating responses for a job or a step, and based on the kind of task a step executes. For a complete list of events for a job or task, see the help topic for the Responses property page for the job or task.

If you want adTempus to take action based on one of these events, you associate a Response with that event.

Related Concepts

Responses, Events, and Actions 279

Reference

Response Event Properties 281
Response Properties 280

Response Event Properties

A Response Event defines an event that will trigger the Response. In the Response Event properties window, select the event that you want to respond to.

The events that are available will depend on the job or kind of task you are creating the response for. For a list and description of the events that are valid for a job or task, see the help topic for the Responses page of the job or task type.

Use a script to determine whether the Response will be activated

Use this option if you want to run a Script to determine wether the Response should execute. adTempus executes the script when the event occurs during job execution. The script then returns a True or False result to indicate whether the Response should execute.

For example, if you have selected the "Step Ends" event, adTempus will run your script when the step ends. Your script then determines whether the Response will execute.

Related Concepts

Events 280

Actions

Actions specify behavior that adTempus should execute when Events trigger Responses.

The following actions are available:

Action	Description
Job Control Action	Used to start, restart, terminate, hold, release, or delete a job or job step.
Job Variable Update	Used to set the value of a Job Variable.

Action	Description
Action	
Notification Action	Used to send notification messages by e-mail, SMS, etc.
File Capture Action	Used to capture and store output files produced by the job.
Script Action	Used to execute a script, allowing you to develop your own action logic.

Related Concepts

Responses, Events, and Actions 279

Reference

Response Properties 280

File Capture Action

File Capture Action

The File Capture action is used to "capture" files produce by your job. Captured files are copied into the adTempus data directory and retained as part of the history for the job.

For example, if a scheduled program creates a log file of its activities each time it runs, you can capture this file so that it is retained with the job's history.

The File Capture action also gives you the option of sending captured files by e-mail. To e-mail files without capturing them in the job history, use the Notification Action or Notification Task instead.

To view captured files, view the properties for the instance once the job or step completes. The files will be found in the Captured Files list for the job; from here you can open the files or copy them to a new location.

ⓘ Captured files are associated with the instance that originally captured them. When that instance is deleted, its captured files are deleted as well. See the job's General page to specify how long the job history should be retained.

Related Concepts

Responses, Events, and Actions 279
Actions 281

Reference

File Capture Action Properties 283

File Capture Action Properties

The **File Capture Action Properties** window contains the settings for a File Capture Action.

Property Pages

Files

The **Files** page lists the files that will be captured.

Files to Capture

Add, edit, or remove File Specifications that define the files to be captured.

Description/Notes

Optionally, enter a description or notes for this action.

Send

The **Send** page allows you to have adTempus e-mail files once they are captured.

If you want to send files without saving them in the job history, use the Notification Action instead.

E-mail captured files to the following recipients

Add, modify, or remove the notification recipients to whom the files should be sent.

Include captured console output

If this option is checked, the e-mail message will also include any captured console output for the step or job invoking the action.

Subject of message

Optionally, provide a subject for the message that will be sent. If you do not specify a subject, a default subject will be used.

Message to send

Optionally, specify the message to be sent. If you do not specify a message, a default message is used.

Notification Severity

Specify the severity (importance) of this message. Notification Recipients, Addresses, and Groups can all be configured to receive only messages that meet specified severity criteria.

Related Concepts

File Capture Action 282

File Specification Properties

Each File Specification defines a file (or set of files) to capture or attach for a File Capture Action, Notification Action, or Notification Task.

File Specification

Specify the file(s) to be captured. You must specify the path and name, but you may use wildcards. For example:

- `c:\proddata\data1.log` looks for a specific file
- `c:\proddata\*.log` looks for any file with a ".log" extension in the `c:\proddata\` directory.

The **File Specification** may contain Job Variables, which will be expanded to resolve the file name.

Include subdirectories

If the **File Specification** contains wildcards and the **Include subdirectories** option is checked, adTempus will check subdirectories of the directory named in the **File Specification**. For example if your **File Specification** is `c:\proddata\*.log`, adTempus will look for a file with the extension ".log" in the `c:\proddata\` directory and all of its subdirectories.

Only capture files with the Archive attribute set

If this option is checked adTempus will only capture files that have the Archive attribute set (typically this indicates that the file has been modified since it was last backed up). Note that adTempus does not reset the Archive attribute after it captures the file.

Only capture files modified since the job started

If this option is checked adTempus will compare the last modification time of the file to the time that the job started. Files that were last modified before the job started will not be captured.

Delete files after they are captured

If this option is checked adTempus will delete the original file(s) after the file(s) have been copied to the adTempus data directory. Otherwise the original files are left alone.

Job Control Action

Job Control Action

The Job Control action is used to execute and manipulate jobs and job steps. It can be used to:

- Run, abort, restart, hold, release, or delete a job
- Run or restart a step (in the current or another job)
- Change the status of the job or step that is executing the action

Either the Job Control Action or the Job Trigger can be used to link jobs together in chains. See the How To: Link Jobs Together topic for a comparison of the two approaches.

Related Concepts

Responses, Events, and Actions 279
Actions 281

Related Topics

Job Condition 263
Job Trigger 236

Reference

Job Control Action Properties 285
How To: Link Jobs Together 494

Job Control Action Properties

The **Job Control Action Properties** window contains the settings for a Job Control Action.

Property Pages

Action

Action to take

Select the action that will be carried out:

Action	Description
Hold a job	Places the target job on hold. Can be used to hold the job that is executing the action, or another action.
Release a job	Releases the target job. Can be used to hold the job that is executing the action, or another action.
Restart the job from the beginning	Restarts the target job from the beginning. Applies only to the job that is executing the action. This results in a new instance of the job being started. Subsequent steps in the original instance are not executed.
Run a job	Runs a job. The job can be started from any step. The new job is executed in parallel with the current job. That is, the Job

Action	Description
	Control action returns control immediately to the caller, without waiting for the target job to complete.
Terminate a job	Terminates (aborts) a running job. When this action is applied to "The current job," it terminates only the instance of the job that is executing the Response. If other instances of the job are running, they are not affected. When this action is applied to another job, it terminates all active instances of the target job.
Restart the step	Restarts the target step. Applies only to the step that is executing the action.
Run another step in the current job	Transfers control to another step in the current job. Once control is transferred, execution continues from the new step. Execution does not return to the original step.
Stop executing steps in this job	Instructs adTempus to stop automatically executing steps in the job. After this action is executed, you are responsible for executing subsequent steps as appropriate, using further Job Control actions.
Set the job's status to failed	Overrides normal success determination rules and sets the job's status to Failed.
Set the job's status to succeeded	Overrides normal success determination rules and sets the job's status to Succeeded.
Set the step's status to failed	Overrides normal success determination rules and sets the step's status to Failed.
Set the step's status to succeeded	Overrides normal success determination rules and sets the step's status to Failed.

Wait __ seconds before restarting

This option is only available when the **Action to take** is "Restart the job from the beginning" or "Restart the step."

Use this option to have adTempus wait a specified number of seconds before it restarts the job or step.

Restart no more than __ times

This option is available only when the action is "Restart the job from the beginning" or "Restart the step."

When this option is selected, adTempus will limit the number of times that the job or step is restarted. Once the limit is exceeded, adTempus will not restart the job or step, and the "Restart limit exceeded" event will be fired.

Set the status of the calling job/step to "Resubmitted"

This option is only available when the **Action to take** is "Restart the job from the beginning" or "Restart the step."

When this option is checked, the current job instance (if the action is "Restart the job") and step have their status set to "Resubmitted." If this option is not checked, they retain their current status.

For example, suppose that you are setting up this action to restart the job if it fails. Instance 1 of the job runs and fails, so the action is invoked and the job restarts, creating instance 2. If the **Set the status...** option is checked, the status of instance 1 is changed from "Failed" to "Resubmitted," making it clear that the restart has occurred, and preventing instance 1 from appearing in the "Failed Jobs" view in the Console. If the **Set the status...**option is not checked, instance 1 retains its original "Failed" status.

Applies To

Determines whether the action applies to the current job (the job that is executing the action), another job on the same server, or a job on another computer. To target another job, you must have "Execute" authority for that job.

If the job is on a different computer, the Console must also be connected to that adTempus server, and the other computer must be defined as a Linked Server in adTempus.

Step

This option is available only when the action is "Run a job" or "Run another step in the current job."

If the action is "Run a job" you may optionally specify the step at which execution should begin. If you do not select a step, the job is executed from the beginning.

If the action is "Run another step in the current job" you must specify the step to execute.

Run only the selected step

If this option is checked, adTempus will run only the step you have selected. Otherwise, adTempus runs the job starting with the selected step and continuing with the remaining steps in the job.

Responses

Choose which Responses you want to execute (this option applies to Responses for the job and for steps within the job):

- **Execute Responses:** All Responses are executed as normal.
- **Do not execute any Responses:** No Responses will be executed.

- **Do not execute any Job Control Responses:**All Responses will be executed *except* Job Control Responses (Responses that run, terminate, hold, or release a job or job step). Use this option if you do not want adTempus to execute any other jobs that are chained to the current job.

Checkpoint

This option is available only when the action is "Run a job" or "Run another step in the current job." You may optionally specify a checkpoint to be passed to the job or step.

Ignore conditions for the job

If this option is checked, adTempus will ignore any conditions that are defined for the target job (this forces the job to execute even if the conditions are not met). Conditions at the step level are not ignored (see next option).

Ignore conditions for individual steps

If this option is checked, adTempus will ignore any conditions that are defined for the target step, or for the steps of the target job (this forces the step(s) to execute even if the conditions are not met).

Force a new instance of the job if necessary

This option overrides the Multiple Instances option for the target job, forcing adTempus to start a new instance of the job even if another instance is already running.

Run only on same Agent

This option is only available if the current server is a Distributed Scheduling Master computer. If this option is checked, the target job will be run only on the same computer as the calling job. If this option is not checked, the job will execute according to the Distributed Scheduling settings for its Queue, and may not execute on the same computer as the calling job.

Run only on Master

This option is only available if the current server is a Distributed Scheduling Master computer. If this option is checked, the target job will be run only on the Master computer; any Agents associated with the job's Queue will be ignored. If the Queue is not configured to run jobs on the Master, the job will not run unless the **Force job to run on Master** option is also checked.

Force job to run on Master

This option is only available if the current server is a Distributed Scheduling Master computer. If this option is checked, the target job will run on the Master computer even if the Queue is not configured to run jobs on the Master.

Variables

The **Variables** page allows you to set or override Job Variables that have been defined for the target job. Values you provide here are used only when the job is executed by this Job Control Action.

For example, you may have a job that is triggered to run independently but can also be started by another job through a Job Control Action. If the job needs to behave differently depending on how it is started, you can set a Variable in the Job Control Action so that the job knows it is being started by another job.

Related Concepts

Job Control Action 284

Job Variable Update Action

Job Variable Update Action

The **Job Variable Update** action allows a job to create or update a Job Variable. The update may affect the current execution of the job only, or can be made permanent.

The same functionality is also available using the Job Variable Task, which can be executed as a Job Step.

Related Concepts

Responses, Events, and Actions 279
Actions 281
Job Variables 88

Related Topics

How To: Set and Retrieve Variables in Scripts 500
Job Variable Condition 269
Job Variable Update Task 178
Predefined Variables 485
Text Edit Window 441

Reference

Job Variable Update Action Properties 289
Job Variable Properties 412

Job Variable Update Action Properties

The **Job Variable Update Action Properties** window contains the settings for a Job Variable Update Action.

Property Pages

General

Name for this action (optional)

Optionally, specify a descriptive name for the action.

Enabled

Uncheck this box to disable the action. If the action is not enabled, it will be skipped at execution.

Description/Notes

Enter any extended descriptive information or notes for this action.

Action

Variable to Update

Enter the name of the Job Variable to update. Click the **...** button to select from a list of already-defined variables. If the variable already exists, the existing value will be updated. Otherwise a new variable will be created.

Note: Do not include percent signs ("%") around the variable name.

Set value to

Select this option to set the variable to a specific value. Enter the value you want to set the variable to.

Check the **Expand variable tokens at runtime** option if you want to expand any variable tokens found in the value you have specified. For example, suppose you are updating a variable named "BaseVariable" and you specifiy that it should be set to `%Variable22%`. If **Expand variable tokens at runtime** is checked, BaseVariable will be set to whatever the value of Variable22 is at runtime. If the option is not checked, BaseVariable will be set to the literal value `%Variable22%` `Variable22` will not be treated as a variable name).

Increase/decrease value by

Select this option to increase or decrease a numeric variable by the amount you specify. If the target variable already exists, it must contain a numeric value type or the update will fail. If the target variable does not exist, it will be created and set to the increment value.

Scope

The **Scope** determines how broadly the variable update will apply.

- **Apply to executing instance only.** The new value of the variable will be seen only within the current instance of the job.

For the remaining settings the new value of the variable will be seen within the current instance of the job, and the definition will also be updated in the variable's containing object, so that future job executions will also be affected.

The user who is editing the settings must have permission to modify the target object or the settings cannot be saved. No permission check is made at job execution.

If auditing is turned on for the affected object, adTempus will generate a change log entry for the variable update. You can change this behavior.

- **Update definition for Step.** The variable is updated in the definition of the Step that is running the update.
- **Update definition for Job.** The variable is updated in the definition of the Job that is running the update.
- **Update definition for Job Group.** The variable is updated in the definition of the Job Group that the executing job belongs to. If the variable is already defined for a group in the job's ancestor hierarchy, the variable will be updated at that level. Otherwise, the variable will be created in the Group that is the job's immediate parent.
- **Update definition for Server.** The variable is updated at the server level.

Related Concepts

Job Variable Update Action 289

Notification Action

Notification Action

The Notification action is used to send notification messages (by e-mail, SMS, etc.) regarding the status of a job. To send a notification message, add a Response to a Job, Job Step, Job Group, or Job Queue.

You can also send notification messages without using Responses by using a Notification Task as a step in your job.

See the Send Notification Messages for Failed Jobs topic for an example of using a Notification Action.

Related Concepts

Responses, Events, and Actions 279
Actions 281

Related Topics

Notification Task 181
Notification Recipients 93
Messaging Setup 417

Notification Recipient 303
Notification Group 309
How To: Send Notification Messages for Failed Jobs 493

Reference

Notification Action Properties 292
How To: Send Notification Messages for Failed Jobs 493

Notification Action Properties

The **Notification Action Properties** window contains the settings for a Notification Action.

Property Pages

Notification

Recipients

Add, edit, or remove the notification recipients who will receive the notification message.

Subject of message

Optionally, provide a subject for the message that will be sent. If you do not specify a subject, a default subject will be used.

The subject may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

Message to send

Optionally, specify the message to be sent. If you do not specify a message, the default message is used.

The default message varies depending on the event that triggered the action. Generally the message contains the job name and instance ID, step number, and a description of the event that triggered the action.

If you specify a message, it will be sent instead of the default message. If you wish to include the default message as part of your message, include the token `%DEFAULTMESSAGE%` in your message. The token will be replaced with the default message.

The message may contain Job Variables, which will be expanded when the message is sent. For more information, see Specifying Notification Subjects, Messages, and Severities at Run-Time.

Notification Severity

Specify the severity (importance) of this message. Notification Recipients, Addresses, and Groups can all be configured to receive only messages that meet specified severity criteria.

★E-Mail Sender

Optionally specify the display name or e-mail address to be used as the sender of e-mail notification messages. If you leave a box empty, adTempus will use the value specified in the SMTP server properties.

You can also override the default values by setting the **NotificationFromName** and/or **NotificationFromAddress**Job Variables for a job, group, queue, step, etc. You can do this either through the properties window for the job, group, etc., or by running a script within the job.

Attachments

The Attachments page allows you to specify files that will be attached to the notification message. In the Files list you can add, edit, or remove File Specifications that define the files to be attached.

ℹ The attached files are not captured and saved as part of the job's history. If you want to save the files, use a File Capture action instead. The File Capture Action can also send the captured files in a notification message.

If you check the **Include captured console output**option, any console output that has been captured for the current step (if this action is being called by a step) or for the entire job (if this action is being called by the job) will be attached to the message.

Related Concepts

Notification Action 291

Script Action

Script Action

The Script Action is used to execute a script. This gives you the flexibility to respond to job events in ways not provided for by the built-in Actions offered by adTempus.

While an Action is being executed, adTempus cannot perform any other "work" on the job with which the action is associated. For example:

- If more than one action has been specified in a response, each action cannot be run until the previous action has completed.
- adTempus cannot start the next step of a job until it finishes the current step. This includes running all actions associated with the step.

Therefore when you use a script action your script should be something that runs quickly, so adTempus can be about its business. If your script goes and launches some external application using the application's automation interface, and then spends 10 minutes doing something with that application, your adTempus job is going to be patiently waiting.

If you want to launch a script without waiting while it executes, you should create a separate job to run the script using a Script Execution Task. Then use a Job Control Action to start that job instead of running a Script Action.

Related Concepts

Responses, Events, and Actions 279
Actions 281

Reference

Script Action Properties 294

Script Action Properties

The **Script Action Properties** window contains the settings for a Script Action.

Optional name for this action

Optionally, specify a descriptive name for the action. Otherwise the name of the target script is used as the description for the action.

Script to Execute

Select or create the script you want to execute.

Description/Notes

Enter any extended descriptive information or notes for this action.

Related Concepts

Script Action 293

Response Example

Job Group

Job Group

Job Groups allow you to group related jobs into "folders" in the adTempus Console. This allows you to better organize your workspace when you have a large number of jobs.

Each group may contain jobs and/or "child" groups. Groups may be nested to as many levels as necessary.

Groups and group hierarchies are global (not personal) settings. That is, when you put a job in a group, the job appears in that group for all adTempus users.

⭐When you delete a group, adTempus will ask you whether you also want to delete all of the jobs within the group. If you choose not to delete the jobs, they will be moved to the next group up.

All jobs in a group inherit the security settings and Job Variables defined for the group.

Aside from security and job variable inheritance, Job Groups are strictly an organizational tool. Placing jobs in the same Group does not create any link or dependency between those jobs. If you need to be able to limit the number of related jobs that are executing at the same time, you can use Job Queues to do this.

Reference

Job Group Properties..295

Job Group Properties

The **Job Group Properties** window is displayed when you create or edit a Job Group. To view or modify the properties for a Group, right-click the group's name in the Console Tree or the Job List and select the **Edit...** or **View...** command.

Property Pages

Group

The **Group** page defines general settings for the group.

Name

Provide a descriptive name for the group (255 characters maximum). The name can be changed at any time. The name must be unique within the parent group.

Description

Enter any extended descriptive information or notes for this group. There is no limit on the length of the text.

Hold Status

The **Hold Status** determines whether and how jobs in this group (and its sub-groups) can be executed.

- **Enabled (not held).** The job can be run by Triggers, Responses, and on-demand. Checking or unchecking this box unchecks or checks the other three options. If this option is checked, the job is fully enabled for all execution types.
- **Hold triggers.** All Triggers defined for the job are ignored. Note: This option holds all triggers for the job, not just Schedule Triggers.

- ★ **Prevent chained execution by other jobs.** Any Job Control Actions that target the job will be prevented from executing it. adTempus will log an error message in the job log for the calling job, indicating that the target job cannot be run
- **Prevent manual execution.** Users will be prevented from executing the job on-demand using the Run command or the adtexec utility.

Cycle ID

Check the **Use this group as a cycle scope** option if you want to be able to assign Cycle IDs for jobs run in this group. Once this option is checked, you must configure a job to update the Cycle ID to mark the beginning of each cycle. See the Cycle ID topic for more information.

The current Cycle ID for the group, if any, will be shown here.

Variables

The **Variables** page defines Job Variables that are inherited by all Jobs and Groups in this group. You may add, modify, or delete Job Variables, or override the values of Variables that the group has inherited.

A Group inherits Variables from its parent group. The top-level (root) group inherits the server-level variables.

Responses

This feature is new for adTempus version 4.

The **Responses** page defines responses that will be executed for all Jobs in the group (and its sub-groups). Configuring Responses at the Group level allows you to easily apply a standard configuration to all jobs. For example, if you want to always send a notification message to the same group of people if a job fails, you can configure a failure Response for the group, rather than configuring a Response for each individual job.

The events that you can configure Responses for at the Group level are the same as those available for jobs.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

Permissions set for the Group are inherited by jobs within the Group.

The following permissions apply to Job Groups:

Permission	Description
Full Control	Permission to perform all actions on the group.
List/Reference	Permission to list this group.
View	Permission to view the properties of the group.

Permission	Description
Create jobs and groups in this group	Permission to create new jobs or groups in this group, or move existing jobs to this group.
Modify	Permission to modify the properties of the group.
Delete	Permission to delete the group.
Administer security	Permission to change the security settings for the group.
Change owner	Permission to take ownership of the group.

Related Concepts

Responses, Events, and Actions 279
Response Example 294

Related Topics

Actions 281
Events 280

Reference

Response Properties 280

Job Queue

Job Queue

Job Queues are used when you need to do any of the following:

- Limit the number of jobs that can be running at the same time.
- Easily hold and release a set of jobs.
- Run jobs on remote computers using the Distributed Scheduling capability of adTempus.

If you do not need either of these features, you can ignore Queues altogether: new jobs are by default assigned to the "Default" Queue, which does not limit job execution in any way.

Job Queues and Job Groups are not related or interconnected in any way: jobs in the same Queue can belong to different Groups, and vice-versa.

Unlike Job Groups, Job Queues do not have a hierarchical structure, because Queues are not used to organize jobs but rather to control their execution.

Limiting Job Execution

In some situations you may need to limit the number of jobs that run at the same time, or prevent certain jobs from running at the same time.

For example, you have three jobs that are CPU-intensive, and allowing more than one of them to run at the same time degrades performance on your server to an unacceptable level. Or you may have a set of jobs that all need exclusive access to a system resource such as a data file or hardware device, so you need to ensure that only one of these jobs is executing at any given time.

A Job Queue allows you to enforce limits like this without the need to set up conditions among the jobs or to rely on careful sequencing of the jobs' schedules. Instead, you assign the related jobs to a single Queue. In the Queue's properties, you can set a limit on the number of jobs from the Queue that can run concurrently.

Whenever a job is triggered (either automatically or through manual submission) it is sent to its assigned Queue for execution. If the Queue's execution limit has already been reached, the new execution request is queued with other pending execution requests, and the job is not executed until all jobs ahead of it in the Queue have been executed. Each time a job finishes, the next job in the Queue is started.

Each Job is assigned a Priority when you configure it. This Priority determines the order in which queued jobs will be executed: when an execution request is queued, the request goes ahead of any other jobs that are already in the queue but have a lower priority.

Holding and Releasing Queues

Like a Job, a Queue can be held to prevent execution of jobs within the queue.

To hold or release a Queue, you can either

- Edit the Queue and change the Hold Status, or
- Right-click the Queue in the Console Tree and use the options on the **Hold** menu.

Distributed Scheduling

When you use the Distributed Scheduling features of adTempus, the Job Queue determines which computer(s) the jobs in the Queue will be executed on. See the Distributed Scheduling Page of the Queue properties for more information.

Job Queue Properties

The **Job Queue Properties** window is displayed when you create or edit a Job Queue. To view or modify the properties for a Queue, right-click the queues's name in the Console Tree and select the **Edit...** or **View...** command.

Property Pages

Queue

Name

Enter a name for this Queue. The name is limited to 255 characters and must be unique.

Hold Status

The **Hold Status** determines whether and how the job can be executed.

- **Enabled (not held).** The job can be run by Triggers, Responses, and on-demand. Checking or unchecking this box unchecks or checks the other three options. If this option is checked, the job is fully enabled for all execution types.
- **Hold triggers.** All Triggers defined for the job are ignored. Note: This option holds all triggers for the job, not just Schedule Triggers.
- **Prevent chained execution by other jobs.** Any Job Control Actions that target the job will be prevented from executing it. adTempus will log an error message in the job log for the calling job, indicating that the target job cannot be run
- **Prevent manual execution.** Users will be prevented from executing the job on-demand using the Run command or the adtexec utility.

Resubmit pending jobs when adTempus restarts

If jobs were queued (awaiting an available execution slot in the queue) when the adTempus service was shut down, adTempus will resubmit those jobs on startup if this option is checked. Otherwise, they will not be resubmitted.

For example, suppose this Queue has an execution limit of 1 (only one job can run at a time). When the adTempus service is shut down, there is one job executing and three more from this queue that have been triggered and are waiting to be executed. If this option is checked, when the service is restarted adTempus will resubmit those three pending jobs to the queue, in the same order they were in at shutdown.

Note that any jobs that were executing at shutdown are not affected by this option, only jobs waiting to be executed.. The restart options for each job determine whether that job will be restarted.

Limit the number of jobs from this queue that can run concurrently

Use this option to limit the number of jobs from this queue that can run at the same time. See the Job Queue Overview topic for more information.

If Distributed Scheduling is in use, this limit applies individually to each computer. For example, if the limit is set to "1" but the Queue is configured to run jobs on two different Agents, each Agent has a limit of one concurrent job, meaning that there may be two jobs (one on each Agent) running in this Queue at any time. If the Queue is configured for Load Balancing, you can additionally limit the number of concurrent jobs against all computers. See the Distributed Scheduling properties for more information.

When a concurrent job has been specified, job execution occurs as follows:

- When a job is triggered, adTempus checks to see if there is a free "execution slot" in the queue. If so, the job begins executing immediately.
- If no slot is available, the job is queued for execution.
- Jobs are arranged in the queue based on their Priority and then the time at which they are submitted to the queue. For example, if a job with priority 200 is triggered and there are already 5 jobs with priority 100 waiting in the queue, the job with priority 200 (higher priority) will execute before the jobs with lower priority. Among jobs with the same priority, jobs execute in the order they were submitted to the queue.
- If a job has Conditions, it does not block the queue while it is waiting for the conditions to be met. For example, if the job with priority 200 is waiting for a File Condition, the lower-priority jobs will execute if they are ready. When a job's conditions are met, it goes back in the queue in its original location (ahead of jobs with lower priority and those with the same priority but triggered later).

Description/Notes

Enter any extended descriptive information or notes for this Queue. There is no limit on the length of the text.

Distributed Scheduling

When you are using the Distributed Scheduling features of adTempus, the **Distributed Scheduling** settings for the Queue determine which computers the jobs assigned to the Queue will be executed on. If the server you are managing is not configured as a Distributed Scheduling Master computer, this page of the Queue's properties will not be visible.

Agent Mode

Select the mode adTempus should use when running the jobs in this Queue.

Limit the number of jobs from this queue that can run concurrently across all target computers

This option is only available if the **Agent Mode** is set to "Load Balance."

Use this option to limit the maximum number of jobs that can run concurrently in this Queue across all computers assigned to this Queue. For example, if the limit is set to "1," only one job at a time will run in this Queue, regardless of how many Agents are assigned to the Queue, or what limit has been set on the **Queue** page.

Run on Master

Check this option to run the job on the Master computer, in addition to any specified agents.

- ⭐ If you select the **Slave** or **Mirror** modes, the job will be run on the Master and on any specified agents. If no agents are specified, the job will run only on the Master.
- If you specify the **Load Balance** mode, adTempus will consider the Master and all specified agents when determining which computer to run the job on.

Run on Remote Agents

Specify the Agents on which the job should run.

When you add or edit an agent, adTempus displays the Remote Execution Options window, which allows you to customize the way the job will be run on this agent.

Evaluate Conditions On

This option is only available if the **Agent Mode** is set to "Slave."

Use this option to specify where job-level conditions for the job should be evaluated (step-level conditions are always evaluated on the computer where the job is actually running).

- **Master:** When the job is triggered, the Master computer evaluates the job-level conditions for the job. If the conditions are met, the execution command is sent to the Agent computers and the job is run. If the conditions are not met, the job is not run.
- **Slave:** When the job is triggered, the Master computer sends the execution commands to the Agents. Each Agent then evaluates the conditions for the job and determines whether the run the job. Thus it is possible that the job will execute on some Agents but not on others.

Variables

The **Variables** page defines Job Variables that are inherited by all Jobs in this Queue. You may add, modify, or delete Job Variables, or override the values of Variables that the Queue has inherited.

A Queue inherits the server-level variables.

Responses

This feature is new for adTempus version 4.

The **Responses** page defines responses that will be executed for all Jobs in the queue.

The events that you can configure Responses for at the Group level are the same as those available for jobs.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

Permissions set for the Queue are inherited by jobs within the Queue.

The following permissions apply to Job Queues:

Permission	Description
Full Control	Permission to perform all actions on the Queue.
List/Reference	Permission to list this Queue.
View	Permission to view the properties of the Queue.
Associate (assign jobs to this queue)	Permission to assign jobs to this Queue.
Modify	Permission to modify the properties of the Queue.
Delete	Permission to delete the Queue.
Administer security	Permission to change the security settings for the Queue.
Change owner	Permission to take ownership of the Queue.

Permission to create new Queues is defined through the Default Queue Security options.

Remote Execution Options

These options determine how jobs from a Queue will be run on the selected Agent. Changing these settings only affects the Agent's behavior for the Queue being edited: if the Agent is also used by other Queues, it may use different settings for those Queues.

To change these settings, select an Agent on the Distributed Scheduling page of the Job Queue's properties, and click the **Edit...**button.

Enable execution on this agent

Check this box to run jobs from this Queue on this Agent. If the option is not checked, the Agent remains associated with the Queue, but jobs from the Queue will not be run on the Agent.

Queue execution commands

This option determines what happens if adTempus cannot deliver an execution command to the Agent (because the Agent is not running or the network connection is broken).

- If this option is checked, adTempus will queue the command. Once the connection is re-established, the command will be delivered to the Agent and the job will be run.
- If this option is not checked, the command will be discarded, and the job will not be run on the Agent.

Notification Recipients

Notification Recipient

A Notification Recipient, or individual, represents a single person or entity who receives notification messages. Each recipient may have any number of addresses, which may be used in different circumstances. For example, an individual may have an e-mail address that receives all notification messages sent to that person, and an SMS (text message) phone number that only receives urgent notification messages.

Each Notification Recipient can be assigned to any number of Notification Groups, which are used simplify notification administration.

Messages are sent to Notification Recipients using Notification Actions, File Capture Actions, and Notification Tasks.

Notification Recipients are managed through the Notification Recipients view in the adTempus Console.

Related Concepts

Notification Recipients Folder 380

Related Topics

Notification Group 309
Notification Task 181
Notification Action 291
Notification Recipients 93
Messaging Setup 417
How To: Send Notification Messages for Failed Jobs 493

Reference

Notification Recipient Properties 304
Notification Address Properties 305

Notification Recipient

A Notification Recipient, or individual, represents a single person or entity who receives notification messages. Each recipient may have any number of addresses, which may be used in different circumstances. For example, an individual may have an e-mail address that receives all notification messages sent to that person, and an SMS (text message) phone number that only receives urgent notification messages.

Each Notification Recipient can be assigned to any number of Notification Groups, which are used simplify notification administration.

Messages are sent to Notification Recipients using Notification Actions, File Capture Actions, and Notification Tasks.

Notification Recipients are managed through the Notification Recipients view in the adTempus Console.

Related Concepts

Notification Recipients Folder 380

Related Topics

Notification Group 309
Notification Task 181
Notification Action 291
Notification Recipients 93
Messaging Setup 417
How To: Send Notification Messages for Failed Jobs 493

Reference

Notification Recipient Properties 304
Notification Address Properties 305

Notification Recipient Properties

The **Notification Recipient Properties** window contains the settings for a Notification Recipient.

Property Pages

Recipient

Name

Provide a unique name for this recipient.

Enable this recipient

If this option is not checked, the recipient can still be used on jobs, but will not receive notification messages until it is enabled.

Severity

Use this option to specify the severity conditions that a notification message must meet in order to be sent to this recipient.

Addresses

Add, modify, or remove addresses for this recipient. The recipient must have at least one address, or it will never receive notification.

When a notification message is sent to a Recipient, the message is sent to each of the recipient's addresses (depending on the selection criteria defined for each address).

Alerts

Check the alert option to have adTempus send this recipient an e-mail message when adTempus encounters a critical problem (such as a problem with the adTempus database). More information.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to notification recipients:

Permission	Description
Full Control	Permission to perform all actions on the recipient.
List/Use	Permission to use the recipient.
View	Permission to view the properties of the recipient.
Modify	Permission to modify the properties of the recipient.
Delete	Permission to delete the recipient.
Administer security	Permission to change the security settings for the recipient.
Change owner	Permission to take ownership of the recipient.

Permission to create new recipients is controlled through the Default Notification Recipient Security options.

Related Concepts

Notification Recipient 303

Reference

Notification Address Properties 305

Notification Address Properties

The **Notification Address Properties** window defines the properties for an e-mail or other notification address for a Notification Recipient. Each recipient may have any number of addresses.

For example, a person may have an e-mail address that goes to her mailbox on the company mail system, and another address for sending messages to her text pager.

For each address, you can use the **Severity** and **Schedule** settings to determine the conditions under which messages will be sent to the address.

Property Pages

Address

The Address page defines the general properties for the address.

Address Type and Address

Select the type of address, and specify the address. The following address types are supported:

Address Type	Description	Recipient Format
SMTP Notification	The recipient is notified using SMTP e-mail messages. Before you use SMTP recipients you must configure SMTP notification.	Specify a valid SMTP e-mail address, e.g., `claire@example.com`.
SMS Notification Using Network Gateway	The message is sent to a mobile device using SMS messaging.	Specify the ID of the device to send the message to. Generally this is the device's telephone number. Depending on your service provider, you may need either the 10-digit or 7-digit telephone number. See the SMS messaging topic for more information.
Jabber Instant Messenger	The message is sent using the Jabber instant messaging protocol.	Specify the Jabber ID or e-mail address of the recipient, e.g., "`example@gmail.com`.

Connection

Choose **Any Available Connection** to have adTempus try all available service providers until it finds one that can deliver the message to this address.

Choose **Use This Connection** to have adTempus use a specific provider for this address.

For an SMS address, you may need to use a specific connection if you are delivering through the carrier's SMS gateway. If you are using a third-party gateway, you can generally allow adTempus to select any available connection.

For an Instant Messaging address type, you must select the service provider that will be used to deliver the message. If no service provider has been defined, you will need to create one through the Notification Setup Window.

Limit messages to

Check this option to limit the length of messages sent to this address. For example, if your text pager has a 300-character limit, you can specify that limit here.

When this option is checked, adTempus will not include any attachments on messages sent to this address.

If you are sending notification to a pager or other SMS device, the maximum message length accepted by the carrier is defined in the Service Provider Properties. The Service Provider Properties also include an option to have adTempus automatically split longer messages into multiple messages so that the entire message can be delivered to you.

In some cases, though, you may not want the full message sent. For example, if you pay for each page or character sent, you may not want a 10-part message delivered to you in its entirety. You can therefore limit the total number of characters that will be sent, which will also limit the number of pages.

The various options work together as follows:

Auto-split option set for service provider?	Message limit specified for address?	Result
Yes	Yes	The message text is first truncated to the limit specified for the address. It is then split into as many messages as necessary to keep each message within the limit specified for the provider.
Yes	No	The message text is split into as many messages as necessary to keep each message within the limit specified for the provider.
No	Yes	The message text is first truncated to the limit specified for the address. If the limit specified for the provider is less, the message is further truncated to stay within that limit.
No	No	The message text is truncated to the limit specified for the service provider.

Severity

Use this option to specify the severity conditions that a notification message must meet in order to be sent to this address.

Schedule

The **Schedule** page allows you to define a schedule that determines when this address will be used. For example, you may want messages sent to your pager only during certain hours.

Override schedule when severity is...

Check this option to override the schedule for messages that meet certain criteria. For example, you may want to receive notification messages on your pager only between 9 AM and 5 PM, except if the message has a severity of 9.

Schedule

Specify as many criteria as necessary. When a notification message is sent to the recipient, adTempus compares the current time to the criteria, and uses an address only if at least one "include" criterion is met and no "exclude" criteria are met.

Related Concepts

Notification Recipient 303

Reference

Notification Recipient Properties 304

Notification Group

A Notification Group is a group of Notification Recipients or other Notification Groups. When a notification message is sent to a group, it is sent to all members of the group (subject to the conditions defined for each member).

By using the schedule option available for each member, you can use a Notification Group to reflect an on-call duty schedule. For example, you may have certain people who handle server problems during the business day and others who are on-call for after-hours problems.

Jobs that you set up don't need to know about the details: they can be configured simply to notify a group. Within the group, you can specify which members should receive messages at which times.

Messages are sent to Notification Groups using Notification Actions, File Capture Actions, and Notification Tasks.

Notification Groups are managed from the Notification Recipients folder in the adTempus Console.

Related Concepts

Notification Recipients Folder 380

Related Topics

Notification Recipient 303
Notification Task 181
Notification Action 291
Notification Recipients 93
Messaging Setup 417
How To: Send Notification Messages for Failed Jobs 493

Reference

Notification Group Properties 309
Notification Group Member Properties 311

Notification Group

A Notification Group is a group of Notification Recipients or other Notification Groups. When a notification message is sent to a group, it is sent to all members of the group (subject to the conditions defined for each member).

By using the schedule option available for each member, you can use a Notification Group to reflect an on-call duty schedule. For example, you may have certain people who handle server problems during the business day and others who are on-call for after-hours problems.

Jobs that you set up don't need to know about the details: they can be configured simply to notify a group. Within the group, you can specify which members should receive messages at which times.

Messages are sent to Notification Groups using Notification Actions, File Capture Actions, and Notification Tasks.

Notification Groups are managed from the Notification Recipients folder in the adTempus Console.

Related Concepts

Notification Recipients Folder 380

Related Topics

Notification Recipient 303
Notification Task 181
Notification Action 291
Notification Recipients 93
Messaging Setup 417
How To: Send Notification Messages for Failed Jobs 493

Reference

Notification Group Properties 309
Notification Group Member Properties 311

Notification Group Properties

The **Notification Group Properties** window contains the settings for a Notification Group.

Property Pages

Group

Name

Enter a unique descriptive name for this group.

Enable this group

If this option is not checked, the group can still be used on jobs, but it will not receive notification messages until it is enabled.

Severity

Use this option to specify the severity conditions that a notification message must meet in order to be sent to this group. Any message that does not meet the condition will not be sent to members of this group.

You can specify severity conditions for an individual member in the properties for that member.

Members

Add, change, or remove group members.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to notification groups:

Permission	Description
Full Control	Permission to perform all actions on the group.
List/Use	Permission to use the group.
View	Permission to view the properties of the group.
Modify	Permission to modify the properties of the group.
Delete	Permission to delete the group.
Administer security	Permission to change the security settings for the group.
Change owner	Permission to take ownership of the group.

Permission to create new groups is controlled through the Default Notification Recipient Security options.

Related Concepts

Notification Group..309

Reference

Notification Group Member Properties 311

Notification Group Member Properties

The **Notification Group Member Properties** window defines the settings for a member of a Notification Group.

Property Pages

Member

The **Member** page contains basic settings for the member.

Settings here only affect messages sent to the member through this group. For example, disabling the recipient here only prevents the recipient from receiving messages through this group.

Recipient

Select the recipient (an individual or another group) who is the member.

Enable this member

If this option is not checked, the member does not receive notification messages sent to the group.

Severity

Use this option to specify the severity conditions that a notification message must meet in order to be sent to this member.

Schedule

The **Schedule** page allows you to define a schedule that determines when this member will be notified.

Override schedule when severity is...

Check this option to override the schedule for messages that meet certain criteria.

Schedule

Specify as many criteria as necessary. When a notification message is sent to the group, adTempus compares the current time to the criteria, and notifies a member only if at least one "include" criterion is met and no "exclude" criteria are met.

Related Concepts

Notification Group 309

Reference

Notification Group Properties 309

Scripts

Using Scripts in adTempus

Scripts are small "programs" that you write in various programming languages to carry out a task or add functionality to adTempus.

Scripts are represented within adTempus by the Script object, which allows you to share scripts among jobs and users, and to restrict the use of scripts with security settings.

Scripts can interact with adTempus using the script integration API to perform actions like getting and setting Job Variable values, setting the Checkpoint for the job, logging messages to the Job Log, and sending e-mail messages.

Kinds of Scripts

There are two broad categories of scripts in adTempus:

- **Task Scripts** are scripts that run as the tasks of Job Steps. Generally the purpose of a task script is to accomplish a task that is external to adTempus.
- **Extension Scripts** are scripts that run to augment or customize the behavior of adTempus.

Task Scripts

You may already have scripts on your computer—such as VBScript (vbs) or PowerShell (ps1) scripts—that you use to perform various processing tasks. You can execute these scripts in adTempus using the Script Execution Task.

We also include Windows batch files in this category, because they are in some ways treated like other scripting languages within adTempus . However, since they are in other ways quite different from other scripting platforms, batch files are run using the Program Execution Task instead of the Script Execution Task.

The following scripting languages are supported for Task Scripts:

- .NET scripting using VB.NET and C#. This exposes the full power of the Microsoft.NET programming framework, without the need for a separate development environment or compiler.
- Windows Script Host (WSH) scripts such as VBScript and JScript.
- Windows PowerShell scripts.
- Windows batch files (run using the Program Execution Task).

Extension Scripts

adTempus lets you use scripts to extend and customize the behavior of adTempus. Scripts can be used:

- As Conditions, to determine whether a job or job step should run.
- As Actions, to perform quick tasks within adTempus.
- In Response Events, to determine when Responses should run.
- To determine whether a job step executed successfully.
- To dynamically determine the command-line parameters for a program to be executed by adTempus.
- To define inline functions, which allow you to call scripts and insert their return values anywhere that Job Variables are supported.

The following scripting languages are supported for Task Scripts:

- .NET scripting using VB.NET and C#. This exposes the full power of the Microsoft.NET programming framework, without the need for a separate development environment or compiler.
- Windows Script Host (WSH) scripts such as VBScript and JScript.

Windows Script Host scripts are supported primarily for backward compatibility. When creating new scripts you are encouraged to use VB.NET or C#, which provide a much more powerful programming environment.

In certain contexts within adTempus the WSH script languages are not supported, and will not appear in the list of available languages. In other contexts scripts written in these languages may not have access to all of the same adTempus context information or features as a .NET script.

Shared Scripts

Shared Scripts are scripts that can be used by more than one job. Any changes made to a Shared Script affect all jobs that use the script.

Shared Scripts are managed through the Shared Scripts view in the adTempus Console. When you create a Shared Script, you can use the security settings to determine which users are allowed to use or modify the script.

Script Libraries

Script Libraries allow you to create libraries of commonly-used code or data that can be shared among scripts in adTempus. This is equivalent to creating a "function library" or "class library" to be used by adTempus scripts.

Script Libraries are managed through the Script Libraries view in the adTempus Console. When you create a Script Library, you can use the security settings to determine which users are allowed to use or modify the script.

Related Concepts

Script 314

Related Topics

Script Properties Window 316
Interacting with adTempus from Scripts 326
Returning a Result From a Script 329
Inline Functions 323
Script Execution Task 199
Script Library 320

Script

A **Script** allows you to store and execute a script (written in a language such as VB.NET, VBScript, or Windows PowerShell) from adTempus.

Scripts can make use of common code stored in Script Libraries and can interact with the adTempus server.

Script Implementation

When you create a new script in adTempus, the Script Editor provides a skeleton script that shows how to structure your script and interact with adTempus. The way that you implement your script differs depending on the scripting technology being used, as discussed in the following sections.

.NET Scripts

A .NET script is implemented as a class that inherits from the UserScriptBase class or one of its specialized subclasses. The base class for your script will depend on the context for which the script is being created. For example, a script created to select files for a File Trigger, the base class will be FileTriggerScriptBase. The skeleton script provided by the Script Editor will derive from the correct base class for the context.

Within the skeleton class provided by the Script Editor, you override the **Run** method to perform the necessary processing and return a result to adTempus. The value that you need to return will depend on the context in which the script is being used; the comments in the skeleton script will provide details.

The methods and properties of a global object named **adTempus** allow the script to interact with the adTempus server to do things like getting Job Variable values and logging messages to the Job Log.

.NET scripts can make use of Script Libraries written in any .NET language (for example, a VB.NET script can reference a Script Libary written in C#) .

.NET scripts can also reference external .NET assemblies that are available on the adTempus server.

Script Parameters

The script base class exposes a Parameters property, which provides information about the current job and step, and provides the information that your script needs to perform its work if it is serving as an Extension Script.

The type of this Parameters object will depend on the context for which the script is being created. For example, when your script devices from FileTriggerScriptBase, the Parameters property will be of type IFileTriggerParameters, which provides information about the files found by the trigger. Your script can then review those files and indicate to adTempus whether it should trigger the job.

Windows Script Host Scripts

Windows Script Host (WSH) scripts (e.g., VBScript, JScript) are implemented as a global code block. You can also create functions within the script.

WSH scripts can only reference Script Libraries that are written using the same language.

The methods and properties of a global object named **adTempus** allow the script to interact with the adTempus server to do things like getting Job Variable values and logging messages to the Job Log.

Script Parameters

adTempus passes parameters to a WSH script through the adTempus.JobVariables collection. For example, if you create a script to evaluate files for a File Trigger, adTempus will pass information about the files to evaluate through Job Variables. The help topic for each adTempus object that can call scripts will indicate the Job Variables that it sets when calling the script.

When creating Extension Scripts you are encouraged to use on of the .NET languages instead of a WSH language, because those scripts provide tighter integration with adTempus.

Windows PowerShell Scripts

Windows PowerShell scripts are only supported as Task Scripts (i.e., as the target of Script Execution Tasks) and cannot be used as Extension Scripts (e.g., to select files for a File Trigger).

When you run a PowerShell script, adTempus automatically inserts an object named "$adTempus" (of type IApplicationIntegration) in the script to support interaction with the adTempus server.

PowerShell scripts can "include" shared PowerShell scripts stored in adTempus. When you include a script, adTempus writes that script to disk at runtime so that your primary script can call it. This allows you to reuse blocks of PowerShell code, similar to the way that Script Libraries work for Scripts.

Related Topics

Script Properties Window 316
Scripts 312
Interacting with adTempus from Scripts 326
Returning a Result From a Script 329
Inline Functions 323
Script Execution Task 199
Script Library 320

Script Properties Window

The **Script Properties Window** contains the settings and code for a script that is stored in adTempus.

Property Pages

Script

Name

Provide a descriptive name for this script. If the script is **Shared**, a name is required and must be unique across all shared scripts. Otherwise the name is optional.

Shared

Check this option to allow the script to be shared among jobs, responses, etc. If the script is not **Shared** it will be available only for the object with which it is associated.

When a script is shared the **Security** page will be visible (see below), allowing you to specify permissions for the script.

Once you have checked the **Shared** option and saved the script, the script cannot be "unshared." If you no longer want it to be shared you must delete it and recreate it.

Description/Notes

Enter any extended descriptive information or notes for this script. There is no limit on the length of the text.

Language

Select the language that the script uses. The following languages are supported by default:

- VBScript
- VB.NET
- C#
- JScript
- Windows PowerShell (only supported in some contexts)
- Windows batch files (only supported in some contexts)

Included Script Libraries

Your script can make use of code defined in Script Libraries. Select any libraries you want to reference here.

The list of available libraries will depend on your chosen script language:

- For a .NET script (VB.NET, C#), you can reference libraries written in any .NET language.
- For a Windows Script Host script (VBScript, JScript), the library must be written using the same language.
- For a Windows PowerShell script, the library must be written using Windows PowerShell.

Included Scripts

For Windows PowerShell scripts, the **Included Scripts** section allows you to reference shared PowerShell scripts. Any scripts that you "include" will be written to a temporary folder (identified by the "ADTJobTemp" job variable) while your script is being executed, allowing you to call those scripts from your primary script.

The **Call Shared Script** drop-down menu in the code editor toolbar will list all scripts available to be referenced. When you select a script from the list, the editor will insert a call to that script at the current cursor location. The following syntax is used to call shared scripts:

```
& ($adTempus.JobVariables["ADTJobTemp"] + "\included script name.ps1")
```

Included Batch Files

For batch files, the **Included Batch Files** section allows you to reference shared batch files. Any batch files that you "include" will be written to a temporary folder (identified by the "ADTJobTemp" job variable) while your batch file is being executed, allowing you to call those batch files from your primary batch file.

The **Call Shared Batch** drop-down menu in the code editor toolbar will list all batch files available to be referenced. When you select a batch file from the list, the editor will insert a call to that batch file at the current cursor location. The following syntax is used to call shared batch files:

```
call "%ADTJobTemp%\included batch file name.cmd"
```

Referenced Assemblies

If you have selected one of the .NET languages (VB.NET or C#), your code may reference other .NET assemblies. The following assembly references are automatically added for all scripts, even though they are not listed here:

- System.dll
- System.Core.dll
- System.Data.dll
- System.Xml.dll
- ArcanaDevelopment.adTempus.ApplicationIntegration.dll
- ArcanaDevelopment.adTempus.ScriptEngine.dll
- ArcanaDevelopment.adTempus.Shared.dll

If your script makes use of any other assemblies, you must explicitly reference them.

Script

Provide the body of the script.

When you initially select the script language, the script body will be filled with a template script for you to use as a starting point. The contents of the template will depend on the context of the script (for example, the template for a Script Execution Task will be different than the template for a Condition Script).

In most cases, your script must return a result to adTempus to indicate the outcome of the script and tell adTempus how to proceed.

Click the new document button on the toolbar to reset the script body to the template for the selected language.

Limit script execution

Specify the maximum time (in minutes) that the script should be allowed to run. (This feature protects against "runaway" scripts.)

If this script is used in a Script Execution Task, which also allows you to specify a time limit, adTempus will use the more restrictive of the maximum run times specified for the Script and the Task.

Run isolated from other scripts

When this option is checked, this script will run in a host process that is not shared by any other scripts. See the Script Security and Isolation topic for more information.

Use 32-bit script host

When this option is checked, the script will be run as a 32-bit process when run on a 64-bit platform. If this option is not checked, scripts (other than Windows Script Host scripts) run as 64-bit processes. Use this option when your script has a dependency on a 32-bit components, driver, etc.

Windows Script Host scripts (VBScript, JScript) always run as 32-bit, regardless of whether this option is checked.

Validate

Click the **Validate** button to validate your script (not available for batch files or PowerShell scripts). adTempus will check to ensure that the code is valid.

Test

Click **Test** to execute your script (not available for batch files). If the script returns a value, that value will be displayed.

Security

The Security page appears only if the script is Shared.

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to scripts:

Permission	Description
Full Control	Permission to perform all actions on the script.
List/Use	Permission to use the script.
View	Permission to view the properties of the script.
Modify	Permission to modify the properties of the script.
Delete	Permission to delete the script.
Administer security	Permission to change the security settings for the script.
Change owner	Permission to take ownership of the script.

Permission to create new scripts is controlled through the Script Security settings.

Related Concepts

Script 314

Related Topics

Scripts 312

Interacting with adTempus from Scripts 326
Returning a Result From a Script 329
Inline Functions 323
Script Execution Task 199
Script Library 320

Script Library

Script Libraries allow you to create libraries of commonly used code or data that can be shared among scripts in adTempus. For example, suppose that you have several jobs where you need to pass the current date as one of the command-line parameters to the scheduled program (but the other parameters are different for each job, so you cannot use a single Shared Script for all of the jobs).

Rather than repeating the necessary formatting code in the command-line script for each task, you can create a function in a Script Library, and then reference the library from the scripts. This in turn allows you to call the functions and subroutines defined in that library.

A Script Library can contain any code or data declarations that are valid for the scripting language you specify for the library.

InlineFunctions Library

The predefined library named "InlineFunctions" contains Inline Function definitions and cannot be deleted. Functions in this library can be used as inline functions, and can also be referenced by scripts just like regular script libraries.

Script Library Code and Namespaces

The various scripting languages all handle namespaces slightly differently, and this may affect the way you are able to define and call code in your script Library.

WSH Scripting Languages

The Windows Scripting Host (WSH) languages (VBScript, JScript, and other WSH-compatible languages) do not support multiple namespaces. As a result, all of the code for a script and for all of the libraries it references end up in the same namespace. You therefore must take care that you do not have variables or procedures in different libraries that have the same name, if those libraries will be used by the same script.

VB.NET

All VB.NET script and library code used by adTempus is placed in the global namespace. Code for each Script Library must be contained by a public Module (by default the Script Library Editor will set the module name to match the library name, but this is not required).

Therefore you may have variables and procedures in different libraries that have the same name, but you may need to qualify a the variable or procedure name using the library name when you use it in a script.

For example, you have created a Script Library named "MyLibrary," and the module defined in it is also named "MyLibrary." Within that module you have a public function named "MyFunction."

When you reference this library from a script, you can call your function using either `MyLibrary.MyFunction` or `MyFunction`.

However, if you create a second library that also contains a function named "MyFunction," and both libraries are referenced by your script, "MyFunction" by itself is ambiguous, and you will need to call `MyLibrary.MyFunction`.

C#

All C# scripts and libraries by are created to use the "UserScript" namespace. You can change the namespace if you wish, and make references across namespaces as necessary. The "adTempus" object used for interaction with adTempus is always in the "UserScript" namespace.

Windows PowerShell

Any function definitions or commands you place in a PowerShell script library are added to the PowerShell runspace before your script is executed, exactly as if they had been included in your script.

Related Concepts

Script 314

Related Topics

Script Properties Window 316
Scripts 312
Interacting with adTempus from Scripts 326
Returning a Result From a Script 329
Inline Functions 323
Script Execution Task 199

Reference

Script Library Properties 321

Script Library Properties

The **Script Library Properties** window contains the settings and code for a Script Library.

Property Pages

Script

Name

Provide a descriptive name for this library. The name must be unique across all scripts, and must contain only letters, numbers, and the underscore ("_") character. No other punctuation or spaces are allowed in the name.

Description/Notes

Enter any extended descriptive information or notes for this library. There is no limit on the length of the text.

Language

Select or enter the language that the script uses.

Referenced Assemblies

If you have selected one of the .NET languages (VB.NET or C#), your code may reference other .NET assemblies.

Included Script Libraries

Your library may reference other Script Libraries in order to call code in those libraries.

Script

Provide any functions, subroutines, constants, etc., that are part of this library. The code you specify here will be available to any scripts that use this library.

When you initially select the script language, the script body will be filled with a template script library for you to use as a starting point.

Validate

Click the **Validate** button to validate your script library. adTempus will check to ensure that the code appears valid.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to script libraries:

Permission	Description
Full Control	Permission to perform all actions on the script library.
List/Use	Permission to use the script library.
View	Permission to view the properties of the script library.
Modify	Permission to modify the properties of the script library.
Delete	Permission to delete the script library.
Administer security	Permission to change the security settings for the script library.
Change owner	Permission to take ownership of the script library.

Permission to create new scripts is controlled through the Script Security settings.

Related Concepts

Script Library..320

Inline Functions

This feature is new for adTempus version 4.

Inline Functions allow you to use calls to script functions anywhere in adTempus that you can use Job Variables. For example, just as you can insert the value of a Job Variable in the command-line parameters for a program, you can call a script function and insert the result.

Defining Inline Functions

All inline functions are defined in a special Script Library named "InlineFunctions." This library defaults to VB.NET as its language, but you can change the language to C# if you prefer (make this change before you start adding functions, or you will have to convert them all to the new language).

To create an inline function, simply add a public method to this library. The library initially contains a few example methods.

Each method must return a value, but the value can be of any type you wish. adTempus calls the returned object's ToString method to get the value to insert.

Calling Inline Functions

You can insert a reference to an inline function anywhere in adTempus where Job Variables are supported. The syntax for inserting an Inline Function is similar to that for inserting a Job Variable. Suppose you have created a function called "GetDate" that returns the current date. You can insert that function in a text field using this syntax:

```
%=GetDate()%
```

If the return type of your function is something other than a string, you can use .NET formatting patterns to format the result when it is inserted, using the same syntax as for

Job Variables. For example, if your GetDate function returns a DateTime object, you can format the date when you insert it:

```
%=GetDate(){yyyy-MM-dd}%
```

However, you could also call the ToString method to do the same thing:

```
%=GetDate().ToString("yyyy-MM-dd")%
```

Inline functions can accept parameters as well. For example, you can create a function that returns only the path part of a filename:

```
Public Shared Function GetPath(filename as string) as
String
       return System.IO.Path.GetDirectoryName(filename)
End Function
```

You can then call that function as follows:

```
%=GetPath("c:\myfiles\somefile.txt")%
```

You can pass the value of a Job Variable to a function. Suppose you have a variable named "TargetFile" that is set to a file name. You can get the path of the current value of the variable using this function call:

```
%=GetPath(TargetFile)%
```

Function calls can also be nested within one another:

```
%=SomeFunction(GetPath("c:\myfiles\somefile.txt"))%
```

Inline Function Evaluation Language

Each inline function call is treated as a block of code that is evaluated in the same language that you have selected for the InlineFunctions library. That is, if the library uses VB.NET, your inline function call must conform to VB.NET language syntax. If the library uses C#, your function calls must use C# syntax.

For simple function calls this difference generally will not matter. However, watch out for details such as string literal syntax.

For example, the samples above all use VB.NET syntax. If your function library is in C#, though, this call will fail:

```
%=GetPath("c:\myfiles\somefile.txt")%
```

It will fail because of the unescaped backslashes in the string, which are not valid in C#. Instead you would need to use one of these formats for the string:

```
%=GetPath(@"c:\myfiles\somefile.txt")%
```

```
%=GetPath("c:\\myfiles\\somefile.txt")%
```

Advanced Inline Scripting

An inline script block is not limited to a single function call: you can include additional script code as well. (Remember that your code must use the language selected for the InlineFunctions library.)

For example, you can combine the results of two functions to produce a single string (this example uses VB.NET) :

```
%=GetPath(TargetFile) & SomeFunction(GetDate().ToString("yyyy-MM-dd"))%
```

You can also call some .NET Framework classes directly. For example:

```
%=System.IO.Path.GetFileNameWithoutExtension(TargetFile)%
```

The following assemblies are supported for direct calls to .NET classes:

- System
- System.Core
- System.Data
- System.Xml

Variable Treatment

When you use a Job Variable within a script block, you can use a variable reference or a variable token.

To use a variable reference, you insert the name of the variable just as you would if the variable were defined in your code:

```
%=SomeFunction(TargetFile)%
```

To combine variables, or to combine variables with literals, you use the appropriate syntax for the chosen scripting language:

```
%=SomeFunction(TargetFile & " - " & FileTag)%
```

(Both "TargetFile" and "FileTag" are Job Variables defined in adTempus.)

You can instead use variable token expansion, just like you would do outside a script block:

```
%=SomeFunction("%TargetFile% - %FileTag%")%
```

When you use this approach, all variable tokens are expanded (replaced with their value) in a preprocessing step, before the script code is evaluated, and the value is inserted directly into the code. Therefore, you must take care that the result is still valid code. For example, if the "FileTag" Job Variable has a line break in it, the generated code would be invalid, and you would get a script compilation error.

Related Concepts

Script 314

Related Topics

Script Properties Window 316
Scripts 312
Interacting with adTempus from Scripts 326
Returning a Result From a Script 329
Script Execution Task 199
Script Library 320

Interacting with adTempus from Scripts

When adTempus runs a script, the script has access to an integration interface that allows the script to exchange information with the adTempus server. For example, the script can retrieve and set Job Variables, set the Checkpoint for the job, and log messages to the Job Log.

Additionally, .NET scripts derive from a base class that exposes a Parameters object, which provides additional context information to the script. See the script overview topic for more information.

The integration interface is exposed to scripts through a global object named "adTempus" ("$adTempus" for PowerShell scripts).

Integration Interface Members

The adTempus object exposes the following properties and methods. For complete information see the Script Integration API Reference.

Properties

Name	Description
Checkpoint	Gets or sets the Checkpoint for the job.

Name	Description
JobID	Gets the unique ID for the job definition.
JobInstanceID	Gets the unique ID for this job instance.
JobVariables	Gets the Job Variables for the job.
StepID	Gets the unique ID for the job step definition.
StepNumber	Gets the step number

Methods

Name	Description
CaptureFile	Overloaded. Captures a file on disk and stores it in the job history.
LogMessage	Logs a message to the Job Log for the job.
ReplaceVariableTokens	Expands variable tokens in a string
SendNotificationMessage	Sends a notification message through the adTempus notification system.
Sleep	Delays execution for the specified number of milliseconds.

Message Logging

Scripts can report progress, problems, or other information to be stored in the job history using the LogMessage method. Log messages can be sent to two destinations:

- **The main Job Log.** Messages appear in the Job Log and the Message Log Query, where you can search for messages based on a message code. Messages are visible in the Job Log as soon as they are reported by the script.

- **The Script Debug Log.** All messages are written to a single log file that is stored as a Captured File in the job history. Because this file is not saved to the history until the script finishes running, these messages cannot be seen anywhere in the Console while the script is still running. You can search for messages using the search captured file content option in the Search tool.

SWhen a script calls LogMessage with the *messageType* parameter set to MessageTypeEnum.Debug, the message is sent to the Debug Log. All other messages are sent to the Job Log.

If your script logs a large number of messages for debugging purposes, you should use MessageTypeEnum.Debug to send them to the Debug Log rather than the Job Log. Reporting a large number of messages to the Job Log fills the adTempus database (each message is stored as a separate row in the database), and it can be difficult to review the messages in the Console.

If your script already creates a log file using its own logging mechanism, you can use the CaptureFile method to attach that file to the job history.

Examples

The following examples demonstrate how to perform basic operations using the integration interface.

Logging a Message

To log a message to the job history (see discussion above), use LogMessage:

```
adTempus.LogMessage(MessageTypeEnum.Error,0,"Failed to
process import file")
```

Getting and Setting Job Variables

A script can retrieve and set Job Variables using the JobVariables collection:

```
myVariable=adTempus.JobVariables("MyVariable")
adTempus.JobVariables("SomeJobVariable")="A value"
```

When you set a JobVariable this way, the change only applies within this job instance; you are not updating the original definition of the variable. If you need to set a value persistently (so it will be seen by other jobs) use the Job Variable Update Task or Job Variable Update Action.

If you want to persistently update a variable based on a value derived in a script, you must do it in two steps. For example, suppose you need to update the "BatchID" variable for the Job Group.

1. Have your script place the value in a temporary variable

   ```
   adTempus.JobVariables("TempValue")="Batch 4098703"
   ```

2. Include a Job Variable Update Task or Action that sets the "BatchID" for the group from the "%TempValue%" variable token.

Related Concepts

Script 314

Related Topics

Script Properties Window 316
Scripts 312
Returning a Result From a Script 329

Inline Functions 323
Script Execution Task 199
Script Library 320

Returning a Result From a Script

Scripts run by adTempus can return a result to indicate their outcome (equivalent to returning an exit code from an application).

The kind of value your script returns will depend on the way that the script is being used. In places where adTempus expects a result from a script, the documentation will indicate the value that should be returned. For example, a Script Condition requires that the result be either True or False. For a Script Execution task you can return a numeric value.

The way a script returns a result depends on the kind of script:

- WSH scripts (VBScript, JScript, etc.) set the global **Result** variable to the appropriate value. For example,

  ```
  Result=True
  ```

- .NET scripts implement a "Run" method that returns a value. The return value from this method is the return value for the script.

Related Concepts

Script 314

Related Topics

Script Properties Window 316
Scripts 312
Interacting with adTempus from Scripts 326
Inline Functions 323
Script Execution Task 199
Script Library 320

Script Security and Isolation

When adTempus runs a script, it launches a separate process (called "[[[Undefined variable Primary.JobHost Name]]].jobhost.exe") to host the script. The host process runs under the user account specified for the job, thus ensuring that the script cannot perform any actions that the user does not have permission for. Running the script in a separate host process also protects the adTempus server from malicious or misbehaving scripts.

Once the script hosting process has been started, adTempus will send additional scripts that need to be run under the same user account to the same instance of the hosting process, so a single instance of the script hosting process may be running several scripts at the same time.

Once the number of active scripts for a single host instance reaches a limit (about 10 scripts by default), adTempus will start another instance of the host process.

Scripts that run under different user accounts are never hosted by the same instance of the hosting process: they are always kept separate.

In some situations you may want to isolate a script from all other scripts. For example, supposed you have a script that loads an external component to perform processing, and this component has a bug that sometimes causes access violation errors. If such an error occurs, it can cause the script host process to "crash," so any other scripts being run by the host process at the same time will also be interrupted.

To avoid this problem you can use the Run isolated from other scripts option in the Script Properties window. When this option is checked, adTempus will always start a new instance of the script hosting process to run this script, and will not send any other scripts to the same host. This keeps the script fully isolated from other scripts.

Remote Agent

Remote Agent

A **Remote Agent** is a remote computer on which jobs managed by the Master computer can be run. See the Distributed Scheduling Overview for more information on Remote Agents.

Before you add a Remote Agent to the Master computer, you must install the adTempus software on the remote computer, and configure adTempus on that computer to run as an Agent. You can then create the Remote Agent definition on the Master, through the Remote Agents folder in the adTempus Console.

Once you have defined Remote Agents, they can be associated with a Job Queue by editing the queue's Distributed Scheduling properties. Jobs assigned to the Queue will then run on the Remote Agent.

Related Concepts

Distributed Scheduling 349

Related Topics

Distributed Scheduling Recoverability 353
Distributed Scheduling Installation 350
Distributed Scheduling Setup and Administration 351

Reference

Remote Agent Properties 331

Remote Agent Properties

The **Remote Agent Properties** defines the settings for a Remote Agent.

Property Pages

Agent

Agent Address

Specify the network name or IP address of the remote computer. The adTempus service must be installed on this computer, and adTempus must be configured to run as an Agent.

Instance

If the Agent is configured to use a named instance of adTempus, specify the instance name.

Port

If the Agent is configured to use a nonstandard port, specify the port.

Optional Name

Optionally, provide a descriptive name for the Agent to be used instead of the address in the user interface.

Enable this agent

Check this box to enable the Agent.

If the Agent is not enabled, no jobs will be run on the agent.

This computer is not in the same domain as the Master

Check this option if the Agent is not in the same domain as the Master (or a domain with a trust relationship with the domain of the Master). In this case you must install a server certificate for on the Agent.

Certificate Thumbprint

If the server certificate used by the Agent is not trusted by the Master computer (e.g., it is a self-signed certificate), enter the certificate thumbprint (available from the Endpoint Configuration Tool) of the Agent's certificate to force the Master to trust the certificate and allow the connection.

Advanced connection settings

Advanced options can be entered to accommodate special connection scenarios. No advanced options are currently defined.

Variables

The **Variables** page allows you to define Job Variables for this Agent. You can add new Variables, or override the values of Variables inherited from the server level.

Variables you define here will be available to all jobs assigned to this Agent. When a job runs on more than one Agent, it receives the appropriate variables on each Agent.

Job Variables can be used to accommodate differences in the file structure or environment between different Agent computers. For example, suppose you have a job that is being mirrored to three different Remote Agents. This job runs program `myprogram.exe`. On two of the Agents, this program is installed in folder `c:\myprogram`, but on the third it is installed in folder `c:\program files\myprogram`. Your job needs to be able to find the program in the correct folder on each of the three computers.

To accomplish this, you would create a Job Variable named "MyProgramPath" for each of the three Remote Agents, and set this variable to the correct path for that Agent. In your job, you would set the target of your Program Execution step to `%MyProgramPath%\myprogram.exe`. When the job is executed, adTempus will replace `%MyProgramPath%` with the correct path on each Agent.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Remote Agents:

Permission	Description
Full Control	Permission to perform all actions on the agent.
List/Use	Permission to schedule jobs to run on this agent.
View	Permission to view the properties of the agent.
Modify	Permission to modify the properties of the agent.
Delete	Permission to delete the agent.
Administer security	Permission to change the security settings for the agent.
Change owner	Permission to take ownership of the agent.

Permission to create new agents is controlled through the Default Remote Agent Security options.

Related Concepts

Remote Agent 330

Security

Security Overview

There are two aspects to security in adTempus:

- **Job execution security** determines what actions an executing job is able to perform.
- **adTempus access security** determines which users are permitted to view or administer data in adTempus, and which actions they can perform within adTempus.

The security framework changed from adTempus 3 to adTempus 4, and some manual configuration of security will be required if you have upgraded from an earlier version. See the Security Changes topic for more information.

Job Execution Security

Each job in adTempus is assigned a Credential Profile (using the **User Account** selector in the job properties), which specifies the Windows user account that the job will run under. When adTempus runs the job, it logs in as the specified user and runs the job in that user's security context.

This means that any programs, scripts, or other tasks run by the job will have the same security limitations as if that user were running the program directly.

The Credential Profile only affects what permissions the job has when it is executed. It does not have any effect on which users are allowed to manage the job within adTempus.

adTempus Access Security

Management of jobs and other objects within adTempus is controlled by the adTempus security framework, which provides granular control over permissions granted to users.

Each adTempus user is represented by a Security Login. When a user attempts to connect to adTempus, adTempus identifies the correct Login based either on the user's Windows identity or on the user ID and password entered by the user.

Each Login is assigned to one or more Security Groups, which allow permissions to be managed easily for sets of users who need similar access.

As in Windows, each secured object support various rights. Individual users or groups are either granted or denied rights. For example, most objects have a "View" right and a "Modify" right. This allows a user to be granted permission to view, but not modify, an object. The property sheet for each secured object contains a Security page, where a standard security editor is used to modify the permissions for the object. adTempus supports security inheritance, meaning that each object inherits permissions from its "parent." See the Security Inheritance topic for more information.

Security Logins and Groups are managed using the Server Security Settings window.

See the Security Configuration Guidelines topic for recommendations on how to configure security and information on initial security setup.

Related Topics

Security Login 339
Security Inheritance 335
Security Group 342
Security Editor 338
Initial Permission Setup 345
Administrator Provisioning 347
Permission Management Guidelines 343

Security Changes from Prior Versions

adTempus 4 has a slightly different security model than previous adTempus versions did. As a result of this, **some manual intervention is required** when upgrading your adTempus installation.

Please review the following information on breaking changes and the steps required to address them.

Administrator Access

In prior versions, members of the Administrators group (in Windows) on the adTempus server were automatically granted Administrator permission in adTempus. No user configuration was required in order for these users to connect to and use adTempus.

Beginning with adTempus 4, there is no automatic access granted to Windows Administrators.

What the upgrade process does: The upgrade process will create an Administrator login for the user who runs the installation/upgrade process and/or for other users designated during the upgrade process. No other users will be configured as Administrators.

What you need to do: Log in under one of the accounts provisioned as an Administrator, then create additional user accounts as necessary (see next section). To make a user an Administrator, add the Login to the "adTempus Administrators" security group.

See the Administrator Provisioning topic if you are unable to log in to adTempus after the upgrade.

Security Logins

In prior versions, there was no need to explicitly define users in adTempus. All users were authenticated automatically using Windows security, and permissions within adTempus were linked directly to the user's Windows identity.

Beginning with adTempus 4, each user must have a Security Login defined in adTempus. Windows authentication can still be used to automatically authenticate users, but a Login must be created in adTempus first.

There is no longer a "Connect" permission to assign to users: a user is permitted to connect if there is a corresponding Security Login.

What the upgrade process does: When the upgrade process encounters a Windows identity that has adTempus permissions mapped to it, it will create an adTempus Login and assign the same permissions.

What you need to do: Log in under using an Administrator login (see previous section), then review the user logins created by the conversion process (in the Server Security Settings window). Create additional logins as necessary to allow additional users to connect. Review and adjust permission assignments as necessary (use the Security Configuration Report to get a listing of all permission assignments).

Security Groups

In prior versions, you could grant permissions (including Connect permission) to Windows security groups. Those permissions then applied to any user who belonged to that group.

Beginning with adTempus 4, Security Groups (or roles) can be defined and managed in adTempus. They are not linked to Windows security groups.

What the upgrade process does: If the upgrade process encounters a Windows security group that has adTempus permissions mapped to it, it will create a corresponding adTempus Security Group with the same permissions. However, it will not assign users to the group.

What you need to do: Review the groups created by the conversion process and assign Logins to the groups as appropriate. Review and adjust permission assignments as necessary (use the Security Configuration Report to get a listing of all permission assignments).

Additional Changes from Version 2

If you are upgrading from adTempus version 2, please review the additional changes between version 2 and version 3.

Security Inheritance

adTempus supports security inheritance, meaning that each object can inherit permissions from its "parent." The model is similar to that of the Windows file system, where each file or folder inherits permissions from the folder above it. This inheritance makes it easy for you to manage security for large groups of users and objects.

Prior to adTempus version 3, security settings in adTempus were not inherited, meaning that each object's security had to be managed individually. If you have upgraded from version 2.x, you may want to update your security settings as described in the Security Changes topic to simplify security administration.

See the Security Editor topic for information on how to control which permissions get inherited.

The following list shows how permissions are inherited. Permissions assigned at each level (or "container") are inherited by all lower levels.

Permission to create a new object of a given type is controlled by the permissions on the container for that object. For example, permission to create a new job is controlled through the security settings of the group in which the job is being created. Permission to create a new Shared Script is controlled through the security settings of the **Scripts** folder.

Container	Notes
Root	The Server Security Settings window provides access to the root server security permissions. Permissions set here are inherited by all objects in adTempus. For example, if you want a set of users to be able to View all objects in adTempus, you would assign View permission here. If you want to grant permissions that apply to all Scripts but not to all Jobs, you need to configure permissions at a lower level
Job Groups	Each Job Group inherits security from the level above it and has its own security settings that affect the group and all groups and jobs below it. To view or edit security settings for a group, right-click the group in the Console Tree, then choose **Edit** and go to the Security page of the group properties window. The **Jobs** node represents the root group, and has security settings just like the other groups. Right-click the Jobs node, choose **Edit**, and go to the Security page of the properties window. Permissions that you assign in the root **Jobs** group are inherited by all groups and jobs in adTempus.
Jobs	Each Job inherits security from its Group and has its own security settings that apply only to itself. To view or edit the security settings for a Job, go to the Security page of the job properties. To simplify permission management, you should when possible set permissions at the Group level rather than configuring security separately for each job.
Queues	Each Job Queue inherits security from the **Queues** node in the Console Tree and has its own security settings. To view or edit security settings for a queue, right-click the queue in the Console Tree, then choose **Edit** and go to the Security page of the properties window. The **Queues** node contains security settings that apply to all queues. To view or edit permissions and control which users are permitted to create Queues, right-click the Queues node and choose **Queue Security**. These settings
Notification Recipients	Each Notification Recipient (individual or group) inherits security from the **Notification Recipients** node in the Console Tree and has its own security settings. To view or edit security settings for a notification recipient, go to the Security page in the property window for the item.

Container	Notes
	To manage the default security for all Notification Recipients and control which users are permitted to create Notification Recipients, right-click the **Notification Recipients** node in the Console Tree and choose **Notification Recipient Security**.
Shared Schedules Holiday Definitions	Each Shared Schedule and Holiday Set inherits security from the Schedules Security and has its own security settings. To view or edit security settings for a schedule or holiday set, go to the Security page in the property window for the item. To manage the default security for all Shared Schedules and Holiday Sets and control which users are permitted to create Shared Schedules and Holiday Sets, right-click the **Shared Schedules** or **Holiday Definitions** node in the Console Tree and choose the **Schedule Security** item. (The same security settings apply to both Shared Schedules and Holiday Sets.)
Shared Scripts Script Libraries	Each Shared Script and Script library inherits security from the **Scripts** node in the Console Tree and has its own security settings. To view or edit security settings for a shared script or script library, go to the Security page in the property window for the item. To manage the default security for all Shared Scripts and Script Libraries and control which users are permitted to create Shared Scripts and Script Libraries, right-click the **Scripts** node in the Console Tree and choose **Script Security**.
Remote Agents	Each Remote Agent inherits security from the Remote Agents node in the Console Tree and has its own security settings. To view or edit security settings for a Remote Agent, go to the Security page in the property window for the item. To manage the default security for all Remote Agents and control which users are permitted to create Remote Agents, right-click the **Remote Agents** node in the Console Tree and choose **Remote Agent Security**.
Messaging Service Providers	Each Messaging Service Provider inherits security from the default Messaging Service Provider security settings and has its own security settings. To view or edit security settings for a provider, go to the Security page in the property window for the item. To manage the default security for all Messaging Service Providers and control which users are permitted to create Messaging Service Providers, click the **Security** button in the Messaging Service Providers Window.
File Service Providers	Each File Service Provider inherits security from the default File Service Provider security settings and has its own security settings. To view or edit security settings for a provider, go to the Security page in the property window for the item. To manage the default security for all File Service Providers and control which

Container	Notes
	users are permitted to create File Service Providers, click the **Security** button in the File Service Providers Window.
Credential Profiles	Each Credential Profile inherits security from the default Credential Profile security settings and has its own security settings. To view or edit security settings for a Credential Profile, go to the Security page in the property window for the item. To manage the default security for all Credential Profiles, click the **Security** button in the Credential Profiles Window.

Related Concepts

Security.. 333

Security Editor

The **Security Editor** is found on the **Security** property page for each object that can be secured in adTempus, and is used to modify the permissions for the object.

To make changes you must have the "Administer Security" right for the object; otherwise you will be able to view but not change security.

The Security Editor is similar to the security properties offered by the Windows Explorer. Using the editor you may add and remove users and groups, and change the permissions for each.

Users and Groups

This list shows all users and groups who have permission for this object. Select an entity to view or change the permissions for that entity.

Permissions inherited from a higher-level object are indicated with "(inherited)" after the name. You cannot modify or remove inherited permissions, but you can override those permissions by adding a new entry for the same user or group and then allowing permissions that were not inherited or Denying permissions that were inherited.

The special login "<Object Creator>" is used to specify the permissions that the creator of an object automatically receives for that object. More information.

Apply Permissions To

This option is only available if the object your are editing is a "security container" (that is, if other objects inherit security from it). For example, a Job Group is a security container because the jobs within that group inherit its security settings.

⭐This option determines whether the settings you are making will apply to the current object only, to the current object and its children, or only to its children.

For example, you want to give a user permission to modify all the jobs within a group, but not the group itself. To do this, you would select the "Jobs only" option before granting the "Modify" permission.

Permission

Select the permissions to Allow or Deny. The list of available permissions will depend on the object your are editing.

Inherit permissions from parent object(s)

When this option is checked, the object inherits permissions from its parent object (for example, a job inherits from the group it belongs to). Any changes made to the parent object's security settings will affect this object.

When this option is not checked, the object no longer inherits permissions, and changes to the parent do not affect the current object.

Remove explicit permissions on all child objects

This option is only available if the object your are editing is a "security container," that is, if other objects inherit security from it. For example, a Job Group is a security container because the jobs within that group inherit its security settings.

When you check this option, adTempus removes any object-specific permissions you have set for any descendents of this object, and forces them to inherit its permissions.

Related Concepts

Security.. 333

Security Login

This feature is new for adTempus version 4.

A Security Login represents a user of adTempus. Each user of adTempus must have an associated Login, which is used to authenticate (identify) the user and to assign permissions for the user.

Logins are managed through the Server Security Settings window.

Previous versions of adTempus used automatic authentication based on the user's Windows identity, and did not use an explicitly-defined Login account in adTempus. When upgrading from previous versions, it will be necessary to create and configure user logins for all users, even if you are using Windows authentication. See the Security Changes topic for more information.

Security Login Properties

General

adTempus supports two methods to authenticate (identify) a user: Windows (integrated or automatic) authentication and adTempus authentication. The authentication type cannot be changed once the Login has been created.

Windows Authentication

With Windows Authentication, the user is not prompted for a user ID and password when connecting to adTempus; adTempus locates the correct Login based on the user's Windows identity.

Windows Authentication can be used when the user will be connecting to adTempus from the same computer where the adTempus service is running, from another computer in the same domain, or from a computer or domain that has a trust relationship with the adTempus server.

When you choose Windows Authentication, set the Login **Name** to the Windows user ID, including the domain name (for example, "domain\userid"). Use the **Select** button to enter and validate the user ID.

You can configure adTempus to authenticate based on Active Directory group membership. See Automatic Authentication below.

adTempus Authentication

With adTempus Authentication, the user is prompted for a user ID and password when connecting to adTempus. This authentication method can be used when a user needs to be able to connect from a computer outside of the server's trust zone, when Windows authentication cannot be used.

When you choose adTempus Authentication, set the Login **Name** to the user ID the user should use, and enter a password for the user.

Users can change their passwords using the **Configuration > Change Password** command in the Console.

Enabled

Uncheck the Enabled box to disable this Login. The user will not be able to log in to adTempus while the account is disabled.

Convert to standard Login

This option appears if the Login is a dynamic Login. See below for more information.

Description

Optionally enter a description or notes for this Login.

Groups

Select the Security Groups that this user belongs to. The "All Users" group cannot be deselected. Users assigned to the "adTempus Administrators" group have full access to and control over all objects in adTempus.

Automatic Login through Group Membership

If you are using Windows authentication and have a large number of users or potential users of adTempus, you may wish to manage security through Active Directory groups (Windows security groups) as was possible in adTempus 3 and earlier, instead of creating individual Logins for each user. For example, you may want to establish an Active Directory group named "adTempus Users" and configure adTempus to grant a specific set of permissions to any user who belongs to that Active Directory group.

To do so, you create a Login in adTempus that is linked to the Active Directory group, by entering the name of the group instead of an individual user name when you create the Login. This is referred to as a **template Login**.

Then assign this Login to the appropriate adTempus security groups and configure permissions as normal.

When a user connects to adTempus using Windows authentication, adTempus will check to see if there is a Login linked to the Windows account. If so, the user is authenticated using that Login.

Otherwise, adTempus will look at all template Logins and check to see if the user is a member of the corresponding Active Directory group. If so, adTempus grants access and permissions to the user based on the template Logins that it matches. adTempus also creates and saves an individual Login for the user, which is referred to as a **dynamic Login**. Like a regular Login, it appears in the list of Logins in the Security Settings window. However, you cannot directly assign permissions or groups to the Login. Instead, each time the user connects to adTempus, the permissions and group memberships are updated based on the template Logins.

Example

You create an Active Directory group named "mydomain\adTempus users".

In adTempus you create a corresponding Login using Windows authentication and setting the Name to "mydomain\adTempus users". This is now a template Login linked to the Active Directory group.

Also in adTempus you assign the "mydomain\adTempus users" Login to the "Read-Only Users" security group that you have previously created.

User Claire runs the adTempus Console and connects to the server for the first time. There is no individual Login defined for Claire, but adTempus determines that she is a member of the "mydomain\adTempus users" Active Directory group, which has a corresponding template Login.

⭐ adTempus creates an individual Login for Claire and copies the "Read-Only Users" group membership from the template Login (and any other group memberships and permissions assigned to that Login).

This automatically-created Login for Claire now appears in the Logins list.

Converting to a Standard Login

A dynamic Login gets updated each time the user connects, to reflect the current permissions and group assignments for the template Logins that the user matches. If you need to customize the permissions for a dynamic Login, you can convert it to a standard Login by clicking the **Convert to standard Login** button. Once you do this, the Login will no longer be updated from the template Logins, and you can manage it just like any other explicitly-created Login.

Related Concepts

Security.......... 333

Security Group

This feature is new for adTempus version 4.

A Security Group (or role) allows you to group together users who need to be given the same permissions. Each user (represented by a Login) can be assigned to any number of Security Groups.

Groups are managed through the Server Security Settings window.

ℹ Previous versions of adTempus used Windows security groups and did not support groups managed within adTempus. When upgrading from previous versions, it will be necessary to assign users to the correct groups. See the Security Changes topic for more information.

adTempus includes two built-in groups:

- The adTempus Users group contains all adTempus users. Permissions you assign to this group will apply to all users.
- The adTempus Administrators group contains users designated as Administrators. Members of this group have full control over data and actions within adTempus.

You should create additional Security Groups based on the roles that users play, and assign permissions to Groups rather than to Logins whenever possible. See the Permission Management Guidelines for more information.

Related Concepts

Security.......... 333

Security Configuration Guidelines

Security Configuration Guidelines

The security inheritance model in adTempus uses the same concepts as the security inheritance in the Windows file system. Therefore many principals for effective Windows security management can be applied to adTempus as well.

This following topics cover basic security setup:

Permission Management Guidelines

Initial Security Setup

Permission Management Guidelines

Security is easier to manage when it is kept simple. The best way to do this is to work with groups (of jobs and other objects, and users) rather than individuals (individual users, jobs, etc.).

User Groups

Security Groups allow you to group users (Security Logins) together based on the permissions they need (and/or the roles they perform). Whenever possible, grant permissions within adTempus to Groups, and then grant permissions to individual users indirectly, by assigning them to groups.

For example, you may have one set of users who are allowed to view and monitor jobs but not to create or modify them, and a second set of users who are allowed to create and modify jobs. To managed this need, you would create two Security Groups (using the Server Security Settings editor):

- The "Job Monitors" group
- The "Job Editors" group

To assign permissions to these groups, right-click the **Jobs** node in the Console tree and choose **Edit** to open its properties.

In Windows, create two more groups, such as "adTempus Read-Only Users" and "adTempus Update Users" and assign the appropriate people to the correct group. Make the two new groups members of the "adTempus Users" group you already created and now, through Windows security inheritance, the people you assign to "adTempus Read-Only users" also inherit the permission to connect to adTempus, without you needing to assign them that permission explicitly within adTempus.

Object Groups

Just as you should grant permissions to groups of users rather than individual users whenever possible, so you should also assign security access to groups of objects rather than individual objects. Practically speaking, you will most often be tailoring permissions to jobs, so following this principal you should try to set permissions based on Job Groups rather than the individual jobs.

If a set of jobs have a related function (e.g., they are all related to one application on your server) it makes sense to put them in the same Job Group. By the same token, they probably have substantially the same security requirements as well. So rather than modifying the security settings for each one of these jobs, you should set the permissions for the Group that contains them instead. Then as new jobs are added to the group, they automatically inherit the permissions from the group. If a few jobs within the group have specialized security requirements, you can customize them either by modifying those jobs individually to override the inherited permissions, or by grouping them into a lower-level group and customizing the permissions for that group.

See the Security Inheritance topic for information about the levels at which security can be configured.

Object Creator

When you are assigning permissions to a security container (e.g., permissions that apply to all Jobs or to all Notification Recipients) you can specify permissions for a special login placeholder named "<Object Creator>". Permissions that you assign to this placeholder are given to the user who creates an object.

For example, in a typical scenario you might want users to be able to view all Notification Recipients, to create new Notification Recipients, and to modify the Notification Recipients that they have created (but not recipients that others have created).

To accomplish this you would use the following combination of permissions in the Notification Recipient Security window:

- Grant View and Create permission to "adTempus Users"
- Grant Update and Delete permissions to "<Object Creator>"

When a user creates a new Notification Recipient, adTempus will create explicit Update and Delete permissions on the recipient for that user. The user will inherit the View permission configured for the security container.

In a new adTempus installation, a default set of "<Object Creator>" permissions are added to the Server Security Settings, so that they apply to all objects in adTempus.

Related Concepts

Security..333

Initial Permission Setup

In a new adTempus installation, only those users provisioned as adTempus Administrators during installation have permission to connect to and use adTempus. Those adTempus Administrators have permission to connect to adTempus, and full control over all objects within adTempus.

To allow other users to use adTempus, you must create and configure adTempus user logins for them, and assign object permissions.

The order in which you perform these actions will depend on the complexity of your security needs (see the Permission Management Guidelines topic). If you expect your security model to be complex, you may want to set up your security groups first so that you can assign logins to the groups as you create each login.

If your security model is simple (e.g., all non-administrator users will have the same permissions), you can simply assign permissions to the adTempus Users security group (which automatically includes all users) once you have created your logins.

If there is no user who has the necessary permission to manage adTempus security, you can give a user Administrator permissions using the Administrator Provisioning tool.

Creating Logins

To configure users,

1. Start the adTempus Console and connect to the adTempus server (you must be logged in to Windows under an account that was provisioned as an adTempus Administrator during installation).
2. Go to the Server Security Settings window (**Configuration > Security Settings**) and go to the **Logins** page.
3. For each user who needs to use adTempus, create a Security Login. Creating a login for the user gives that user permission to log in to adTempus. Assign the Login to the appropriate Security Groups, if you have created them.

Assigning Permissions

As discussed in the Permission Management Guidelines topic, permissions should be assigned based on groups of users and objects whenever possible (rather than assigned directly to individual users or objects).

Once you have determined your security needs, you should create the appropriate Security Groups and assign permissions to them. Then assign logins to security groups to grant permissions to users.

Default Permissions

In a new adTempus installation, the following default permissions are present:

- Members of the adTempus Administrators group have full control within adTempus. This permission cannot be removed.
- All users have View permission for the "Default" Job Queue. This permission cannot be removed.
- All users have View permission for the root Job Group. This permission cannot be removed.
- All users have View permission for all objects in adTempus. This permission can be modified through the **Security Settings** page of the Server Security Settings window.
- The creator of an object receives permission to view, modify, delete, and execute (for jobs) the object. These permissions can be modified through the Security Settings page of the Server Security Settings window. This is assigned through the "<Object Creator>" placeholder.

Modifying Permissions

By default non-administrator users do not have permission to create any objects in adTempus or to execute any jobs. For those users to do anything other than view data, you must grant permissions to allow them to create or modify jobs, notification recipients, etc.

Permissions for All Objects

To grant a group of users permissions for all objects in adTempus, use the **Security Settings** page of the Server Security Settings window and select the appropriate option for the **Apply Permissions To** setting.

For example, if you want to create a Power Users group that has permission to create , view, and modify all objects in adTempus. After creating the Security Group, you would add it to the access list on the **Security Settings** page, with permissions applied to "All objects below this level," and check all permissions you wish to grant to the Power Users.

By using the "All objects below this level" option, you are preventing those users from being able to change server-level settings and security, so they still don't have full control over adTempus.

Permission to Create Jobs

If you want to give a group of users permission to create jobs, edit the root Job Group (the "Jobs" folder in the Console Tree) or whichever Job Group you want the users to be able to create jobs in. On the Security page of the Job Group Properties, add the group to the access list and grant the "Create jobs and groups in this Group" permission, applying the permission to "Group, Subgroups."

This will give the users permission to create jobs. The default Object Creator permissions will be assigned to each job they create, giving them Modify permission for their own jobs.

These users will not be able to create other top-level object types (Shared Schedules, Notification Recipients, etc.) unless permissions are also granted for those objects. They will be able to create steps, tasks, conditions, triggers, responses, etc., within their jobs.

Permission to Create Other Objects

To give users permission to create other types of objects, you must grant them Create permission on the appropriate security container. See the Security Inheritance topic for a list of those security containers.

For example, suppose you want to allow all users to create Notification Recipients for use on jobs. To do so:

1. Right-click the **Notification Recipients** folder in the Console Tree and choose **Notification Security...** to open the Notification Recipient Security window.
2. In the access list, click the **Add** button and add the "adTempus Users" group to the list.
3. With the "adTempus Users" entry selected, check the Allow Create box to assign the Create permission.
4. Click **OK** to save the changes.

Note that you only assigned the Create permission for all users; you did not select any other permissions. This is because the creator of an object is automatically assigned permission to view, modify, use, and delete that object (this happens through the "<Object Creator>" permissions discussed above), and the default settings give all users permission to view all Notification Recipients.

With this combination of settings, the users can create and manage their own Notification Recipients, but cannot modify Notification Recipients created by other users.

If you had checked the "Update" permission in addition to "Create," you would be granting all users permission to modify all Notification Recipients, whether they are the creator or not.

Related Concepts

Security.. 333

Administrator Provisioning

In order to use adTempus, each user must have a Security Login defined in adTempus. Unlike in adTempus version 3 and earlier, members of the Administrators group are not automatically granted access to or Administrator permission in adTempus.

Creating Administrators During Installation

During installation, the Database Configuration Wizard will prompt you to specify the users who should initially be made Administrators of adTempus. Once setup completes, these users will be able to manage adTempus and add additional users.

Creating Administrators After Installation

If you are an adTempus Administrator, you can add additional Administrators through the adTempus Console as discussed in the security setup topic. If no users are able to log in to adTempus as Administrators, you can use the **Administrator Provisioning** tool found in the adTempus group on the Start menu to give Administrator permission to users.

Enter the name(s) of the Windows users to designate as Administrators. Include the domain name if necessary, and separate entries with semicolons or new lines. For example:

```
ExampleDomain\Bob; ExampleDomain\Cindy
```

After you click **Save** you must restart the adTempus service for the logins to be created.

In order to run this tool you must have permission to update the Registry on the adTempus computer.

If at least one user is still able to connect to adTempus under a login designated as an adTempus administrator or security administrator, there is no need to use the administrator provisioning tool: you can configure logins through the Security Settings window in the adTempus Console.

Related Concepts

Security.. 333

Distributed Scheduling

Distributed Scheduling Overview

The adTempus **Distributed Scheduling** feature allows you to manage jobs on a single **Master** instance of adTempus but execute the jobs on remote **Agent** computers that also have adTempus installed.

With Distributed Scheduling, users connect to the Master adTempus instance to manage jobs, and specify which Agents the jobs should run on. The Master then sends the jobs to the Agents for execution.

- The Master computer can run jobs on itself and on any number of remote agents.
- An Agent is controlled by a single Master. The Agent runs jobs sent to it by the Master. Depending on your license and configuration, and Agent may also run jobs that are managed locally. That is, you can connect directly to the Agent using the Console and schedule jobs that run independently of the Master.

With Distributed Scheduling enabled, each Job Queue has options that determine how and where jobs assigned to that Queue will execute. Jobs can be executed on any combination of the Master and Agent computers.

Although jobs are sent to the Agents for execution, they are still managed through the Master, and the status and history for jobs running on Agents are reported back to the Master. Thus the Console shows the status and history of a job on all the Agents that it runs on. Jobs sent to an Agent from the Master cannot be modified on the Agent.

Distributed Scheduling Modes

Each Queue executes its jobs in one of three Distributed Scheduling **modes**: Mirror, Slave, or Load Balancing.

Note that the mode is set for a particular Queue and affects all jobs assigned to that Queue. Each Agent can run jobs from any number of Queues using any of the modes (that is, an Agent is not limited to executing in a single Distributed Scheduling mode).

Mirror

In Mirror mode, the job is copied to all of the Agents attached to the Queue. The trigger(s) for the job are then evaluated separately on each agent, so the job operates completely independently on each Agent. This mode is essentially equivalent to replicating jobs to the remote Agent. Since the jobs run completely independently of the Master, they are able to execute even if the Master adTempus instance is not running, or the network connection between the Master and Agent is lost.

Changes made to the job on the Master are sent to the affected Agents, and status and history changes from the Agents are sent back to the Master.

Slave

In Slave mode, the trigger(s) for the job are evaluated only on the Master. When the job is triggered, the Master sends the job to each Agent for execution. In this mode, jobs are pushed out to the Agent at execution time, so the Master computer must be running. If an Agent is not available at execution time, the Master can queue the execution command to be sent to the Agent when communications are re-established.

Load Balancing

In Load Balancing mode, the trigger(s) for the job are evaluated only on the Master. When the job is triggered, adTempus runs the job on only one of the computers (Master or Agents) assigned to the Queue. It selects the best computer based on the current load on each computer.

adTempus compares the load based on the criteria listed below, checking them in the order listed. If all available computers have the same load (within a tolerance range) for one criterion, adTempus moves on to the next, and so on, until a "best" target is found.

Evaluation criteria:

- Current number of executing jobs vs. job limit set for the queue
- Average CPU usage for the machine
- Average memory usage for all processes on the computer (% of max)
- Average adTempus load (active jobs and adTempus CPU utilization)
- Average memory available on the computer
- Average queue load (number of active and pending jobs for queue)
- Agent priority

Averages are calculated over the previous 10 minutes with 5-second sampling intervals.

Related Topics

Distributed Scheduling Recoverability 353
Distributed Scheduling Installation 350
Distributed Scheduling Setup and Administration 351
Remote Agent 330

Distributed Scheduling Installation

To use Distributed Scheduling, you must install adTempus on the Master and on each Agent (a separate license must be purchased for each computer). You install the same software on each computer. During installation you choose whether the computer will be a Master or an Agent by specifying the Engine Mode. You can also change this setting after installation using the Engine Mode utility.

It does not matter which order you install the instances, but you must have one Master computer and one or more Agent computers.

Aside from selecting the Engine Mode during or after installation, no other special installation steps are required.

System Options on the Agent

Agents receive their system option settings from the Master. There is therefore no need to configure notification, history retention, etc., on the Agents.

Configuration

Once the software is installed, you are ready to establish connections to the Agents and begin scheduling jobs. See the Setup and Administration topic for details.

Related Concepts

Distributed Scheduling 349

Related Topics

Distributed Scheduling Recoverability 353
Distributed Scheduling Setup and Administration 351
Remote Agent 330

Distributed Scheduling Setup and Administration

Once the adTempus software is installed and running on the Master and Agent computers (see the Distributed Scheduling Installation topic), you are ready to begin configuring and managing Distributed Scheduling jobs.

All Distributed Scheduling configuration is performed through the Master computer, so you should start the adTempus Console and connect to the Master.

Defining Agents

The first step in Distributed Scheduling setup is to tell the Master about the Agents that you want it to manage. Before you proceed, make sure the adTempus software is installed on the Agent computers.

Next, go to the Remote Agents folder in the Console and add a new Remote Agent definition for each of the Agent computers you want to manage from this Master.

Note that a Master can control any number of Agents, but each Agent can belong to only one Master. Once a Master has connected to an Agent, the Agent will not accept connections from any other Masters unless you reset the agent's master using the Engine Mode utility.

Configuring Queues

Once you have defined your Remote Agents, you can begin assigning them to Queues. See the Distributed Scheduling Page of the Queue properties for more information on configuring Queues for Distributed Scheduling, and the Mirror jobs to Remote Agents topic for a setup example.

Configuring Jobs

Once a Queue is configured for Distributed Scheduling, any Job assigned to that Queue is executed according to the Queue's Distributed Scheduling settings. You do not need to change any settings in the job itself to make it use Distributed Scheduling.

All changes you make to the job will be sent to the Agents as necessary.

However, you should review each Distributed Job's settings to make sure that the job will execute properly on all computers assigned to the Queue.

- Any programs, batch files, etc., run by the Job must be installed and configured on each computer.
- The user account used for the job must be valid on each computer.
- Any file paths must be the same on all computers, or you must use Job Variables to ensure that the correct path is used for each Agent. See the discussion in the Remote Agent Job Variables topic for more information.

Administering Distributed Jobs

The following tools in the Jobs View of the Console will help you administer distributed jobs.

- When you select a job, an "Agents" tab is available at the bottom of the view. This tab shows the job's status for each agent that it targets. From this agent list you can also manually run a job on a specific agent only.
- The History and Job Log tabs for a job list instances and events for all agents, and indicate which computer each instance is executing on.
- If you want to manually start a job on the Master or on all Agents, select the job from the Job List and use the **Run** command from the pop-up menu.
- If you want to manually start a job only on one particular Agent, select the job, then go to the Agents tab and select the desired Agent. Right-click the agent name and select the**Run**command from the pop-up menu.

Related Concepts

Distributed Scheduling 349

Related Topics

Distributed Scheduling Recoverability 353
Distributed Scheduling Installation 350
Remote Agent 330

Distributed Scheduling Recoverability

Communication

When the adTempus Master service starts, it attempts to connect to all defined agents (except those that have been disabled).

If no connection can be made, or if a connection is made and subsequently lost, adTempus will retry the connection approximately every 30 seconds until the connection is re-established.

Queued Commands

If the connection between the Master and an Agent fails, both the Master and the Agent will queue any data and commands that need to be transmitted, and will exchange this data when the connection is re-established.

In some cases, you may want adTempus to just "skip" an agent if it is not connected when a job needs to be run. This can be controlled when you link the agent to the queue.

Independent Operation

If an Agent loses the connection to its Master, this has no effect on any jobs that are already executing, as the Agent has a complete copy of the job's data stored locally.

If the job uses the Slave or Load Balancing agent modes, no new instances of the job will be run until the connection is re-established, as the jobs are triggered only by the Master.

If the job uses Mirrored mode, the Agent will continue to execute the job according to its defined schedule (or other trigger) regardless of whether the Agent is connected to the Master.

Related Concepts

Distributed Scheduling 349

Managing Multiple Environments

Managing Multiple Environments

It is common to migrate business applications and/or the jobs that execute them through several stages or environments, such as development, test, staging, and production. This section discusses approaches and recommendations for managing this lifecycle in adTempus.

In a staged deployment model, applications and jobs generally start out in the "development" environment where they can be built, configured and tested without affecting the production environment. They are then migrated (copied) to the next stage or environment, which could be one or more intermediate levels of testing, or could be the production environment. The number of stages in the process does not matter for purposes of this discussion.

In adTempus, you can use a separate instance of the adTempus server (service) for each environment (these instances can be on the same computer or separate computers, depending on how you have your environments set up).

Jobs and other data are moved between environments using the Export/Import tool in adTempus.

The topics walk through the steps required to set up adTempus to support multiple environments:

1. Set up the adTempus instances (software installation)
2. Configure basic settings and reference data in your first environment
3. Copy basic settings and reference data to other environments
4. Optimize job configuration for multiple environments
5. Migrate jobs between environments

Related Topics

Managing Multiple Environments with Separate Instances 354
Configuring Jobs for Multiple Environments 359
Migrating Jobs Between Environments 361
Propagating Initial Setup 358
Initial Configuration 356

Managing Multiple Environments with Separate Instances

Each of your software environments will be represented by an instance of the adTempus server, so that each environment is isolated from all the others. An "instance" of adTempus refers to an instance of the adTempus server (service) software.

Environment Architecture

Depending on your needs and hardware resources, your environments may be implemented using either of these approaches, or a combination of the two:

All environments on one computer

For some implementations it may be feasible to host some or all of your environments on one computer, using separate folders on disk or some other mechanism to maintain different versions of the applications that adTempus is running.

Prior to adTempus 4 some users accommodated this model by maintaining multiple copies of each job in adTempus, organized under Job Groups representing each environment. This approach has many limitations and drawbacks, and is no longer recommended.

Beginning with adTempus 4, you can run multiple instances of the adTempus server on a single computer (all sharing a single license). Therefore each of your environments can be hosted in a separate adTempus instance, isolated from the others.

The remainder of the discussion of multiple environments assumes you will use separate instances for your environments.

Each environment hosted on a separate computer

If each of your environments is on a separate computer, you will use a separate instance (installation) of adTempus on each computer to support that computer's environment.

Installing adTempus Instances

To support your environments, you will install an instance of the adTempus server for each environment that you need to support. For example, if you have development, test, and production environments, you will have development, test, and production instances of adTempus.

Separate Computers

If your environments are on separate computers, install the adTempus server software on each of those computers. Each of these adTempus instances will require its own software license.

Same Computer

If some or all of your environments are on the same computer, install the adTempus server on that computer. Then use the Instance Management tool to create an adTempus instance for each of your environments. All instances on the computer will use the same installation of the software, but the adTempus server service will run separately for each instance.

When you create a new instance of adTempus using the Instance Management tool, you give the instance a name, which is also used when you connect to the instance using the Console. Your instance name should match the environment name. For example, you would create "development," "test," and "production" instances.

When the software is installed, a default (unnamed) instance is automatically created. To avoid confusion, we recommend that you not use this instance. Instead, create a named instance for each environment. Then disable the default instance so it cannot be used accidentally.

To disable the default instance, go to the Services tool in Windows and stop the adTempus service. Then change its startup type to "Disabled."

Considerations for Distributed Scheduling

If you are using Distributed Scheduling to execute jobs on multiple computers, you will need to have separate instances for your Master and for some or all of your Agents as well. For example, if you have a Master and two Agents, you would need to create a Master for development, test, and production environments. If you want to be able to test the distributed scheduling component, you would also need at least one Agent for each environment.

It is not necessary to have the same number of Agents in each environment, or even to use Agents in all of your environments. For example if you use four Agents to load balance your jobs in production, you may configure your Queue to only use two Agents in your test environment, and in your development environment you could configure the Queue to only execute jobs locally (not using any Agents).

Related Concepts

Managing Multiple Environments....................354

Initial Configuration

Begin by identifying your "stage 1" environment, which generally will be your "development" environment. All reference data and jobs will be created here and then migrated to the other environments.

As you begin configuring adTempus, it is important that you create all your jobs and reference objects (shared scripts, notification recipients, etc.) in a single environment and then copy them to the other environments, rather than creating them separately in each environment.

This is because each object in adTempus (each Job, Group, Notification Recipient, Credential Profile, Script, etc.), has a unique internal identifier value that adTempus uses to keep track of the object. This identifier is not the same as the name of the object (because the name can be changed). When you create the objects in one environment and copy them to another, they maintain this internal identifier, which adTempus will use to be sure the correct object gets updated when you copy subsequent changes from one environment to another. If you create the objects separately in two environments, adTempus will not see them as being the same, even if the name is the same.

Background: Kinds of Reference Data

There are two kinds of reference data used by jobs in adTempus:

- **Linked reference data** refers to items such as Shared Scripts, Schedules, Credential Profiles, Holiday Definitions, etc., where a job references a specific item. For example, a job uses a specific Credential Profile.
- **Unlinked reference data** refers to items that are not explicitly referenced by a job. For example, when a job sends a notification message it needs the settings from an SMTP Server definition. But the job doesn't reference a specific SMTP server; it just uses any of the servers defined in adTempus.

This distinction is important because the export/import process used to migrate jobs between environments will automatically copy linked reference data if necessary. But it won't copy unlinked reference data, so that needs to be copied separately, as discussed in Propagating Initial Setup.

Basic Settings

Begin by configuring basic adTempus settings in the General Server Options window. You can then copy these settings to the other environments to serve as a starting point, even though you may wish to change some of them in other environments.

As discussed in the "Configuring Jobs for Multiple Environments" on page 359 topic, you should use Job Variables to define values that need to change when jobs move between environments (such as file paths, server names, etc.). You should define the variables at the server level so that they are available throughout the instance.

Notification Options

Configure any necessary servers/providers in the Notification Options window. For example, configure the SMTP server(s) used to send e-mail messages.

Linked Reference Data

You can configure linked reference data now and copy it over to the other servers, but this is not necessary, as this data will be copied when it is referenced by a job.

Linked reference data includes:

- Holiday Definitions
- Shared Schedules
- Shared Scripts
- Script Libraries
- Credential Profiles
- Notification Recipients (groups and individuals)

Security

Review your security (access) needs for adTempus as discussed in the Initial Security Setup and Permission Management Guidelines topics.

Set up any security groups and logins that you need.

Some additional recommendations apply to make it easier to manage security across multiple environments:

- Assign permissions to Security Groups rather than individual logins. That way you can change permissions on each server by managing group membership rather than having to edit each object to specify which users have permission on which servers.
- Avoid assigning permissions to individual jobs, as these permissions may have to get adjusted each time you copy the job to a new environment. Instead, assign the permissions to Job Groups so that they apply to all jobs within a group. This way, if the jobs need different permissions on different servers, you can adjust the security settings for the groups once (the first time they are copied) and then not update them with changes from other environments when copying data.

See the Importing Security Settings topic for more information.

Related Concepts

Managing Multiple Environments 354

Related Topics

Managing Multiple Environments with Separate Instances 354
Configuring Jobs for Multiple Environments 359
Migrating Jobs Between Environments 361
Propagating Initial Setup 358

Propagating Initial Setup

Once you have completed initial setup of reference data, you need to copy those settings to your additional environment(s). This is done using the Import/Export tools in the adTempus Console. You will copy the data from the *source* instance to one or more *target* instances.

1. Make sure the Console is connected to both the source and target instance of adTempus.
2. Select the source server in the Console Tree.
3. Go to **Tools > Import/Export > Export** to open the "Data Export" on page 445.
4. In the **Data Export** window, review the following categories (available in the **Show Objects of Type** selector) and select each reference item that needs to be copied:
    - Messaging Service Providers
    - File Service Providers

    In the "Server Settings" category, check the boxes for all available settings types.

Note: You generally do not need to copy objects from other categories, as they are linked reference objects that will automatically be copied over later if they are referenced by jobs. However, if you want to have different settings for these objects in different environments, you can copy them over now and make the required changes.

5. In the **Export To** section, select the Server option and choose the target server from the list.
6. Click **Export**. The Import Wizard will open to import the data into the target server.
7. Click **Next** to go to the **Select Objects** page.
8. Select all of the objects and click **Next** to go to the **Import Options** page.
9. On the **Import Options** page, check the following options that are not checked by default:
    - **Import security settings for objects**. This is necessary to replicate any security scheme you have set up.
    - **Save all imported objects if no problems are found**
10. Continue through the wizard to complete the import on the new server.

Related Concepts

Managing Multiple Environments 354

Related Topics

Managing Multiple Environments with Separate Instances 354
Configuring Jobs for Multiple Environments 359
Migrating Jobs Between Environments 361
Initial Configuration 356

Configuring Jobs for Multiple Environments

When a job is migrated from one environment to the next, the goal is that no changes should be required to the job settings, as this would allow for the introduction of errors into the configuration. There are several job configuration strategies that can help you avoid the need for changes.

Use Job Variables

Wherever you have a value (such as a path or file name) that may need to change between environments, define a Job Variable for that value, and use the variable in the job. In each environment you can give the variable the value that is appropriate for that environment, so when the job moves to the new environment it will use the new value.

You can define the variables at the server level so that they are available throughout the instance. Variables defined at this level do not get copied in a normal export, so you don't need to worry about them accidentally overwriting the values in another environment.

There are other Job Variables that you will want to have set the same in all environments (you will want the changes to be migrated between environments). These variables should be defined in the root Job group (by editing the "Jobs" node in the Console Tree) or at some lower level.

Use Notification Groups

If your jobs send notification messages, and you want the messages to go to different recipients in different environments, your jobs should target Notification Groups rather than individual Notification Recipients. The Notification Group can then be configured separately in each environment to notify the appropriate recipients.

This is important because when you migrate a job from one adTempus instance to another using Export/Import, the settings for the job will include the links to the Notification Groups targeted by the job, but not the contents of the group, so making changes in one environment need not affect the others. On the other hand, if the job targets specific recipients, it will also target those recipients in the new environment, and this cannot be changed without modifying the job.

Use the same Credential Profile, even if the user ID is different between environments

When you migrate a job from one instance to another, the job will contain a link to the Credential Profile that the job uses, but this link is based on an internal identifier that is not related to the actual user ID. Therefore the "same" Credential Profile can have different user credentials in different environments.

Suppose that when you run your jobs in your test environment, they need to run under the user account "testuser," but in production they need to run under account "produser." To accomplish this you first create the Credential Profile in your test environment, using the "testuser" account. Next you export this Credential Profile to the production environment so that the profile in the production environment has the same internal identifier as the test profile. Finally, you edit the Credential Profile in the production environment and change the credentials to use the "produser" account. Now any jobs that reference this profile will use the appropriate credentials for the environment.

Related Concepts

Managing Multiple Environments..354

Related Topics

Managing Multiple Environments with Separate Instances..................................354

Migrating Jobs Between Environments 361
Propagating Initial Setup 358
Initial Configuration 356

Migrating Jobs Between Environments

To copy ("promote") jobs from one environment to another, you use the "Data Export" on page 445. This tool can export jobs and other objects directly from one server to another, without using an export file.

Once you have completed initial propagation of reference data for your servers, you generally will only need to explicitly export jobs between servers, and not other objects such as Job Groups, Credential Profiles, Notification Recipients, etc. This is because exporting a job automatically includes any reference objects that the job links to.

To export data from the *source* server to the *target* server:

1. Make sure the Console is connected to both the source and target instance of adTempus.
2. Select the source server in the Console Tree.
3. Go to **Tools > Import/Export > Export** to open the "Data Export" on page 445.
4. In the Export window, select the job(s) that you want to copy (promote).
5. In the **Export To** section, select the Server option and choose the target server from the list.
6. Click **Export**. The Import Wizard will open to import the data into the target server.
7. Click **Next** to go to the **Select Objects** page.
8. On the **Select Objects** page, select the objects you want to import, as discussed in Selecting Objects for Import below. Click **Next** to continue to the **Import Options** page.
9. On the **Import Options** page, decide whether to import security settings, as discussed in Importing Security Settings below.
10. Continue through the wizard to complete the import and save the imported data.

Selecting Objects for Import

On the **Select Objects** page, the Don't import referenced objects that already exist on this server option will be checked by default, and this is generally the setting you want to use when promoting jobs, except when you need to copy modified reference data. With this option checked,

- The jobs you selected in the Export window will be listed and can be imported. If they already exist on the target server, the import will replace their settings
- For any objects that the job references (such as Credential Profiles, Shared Schedules, Notification Recipients),

- If the object does not exist on the target server, it will be imported.
- If the object already exists on the target server, it will not appear in the selection list, and **will not be imported**.

For example, suppose you have created job "Job A" in the Development environment and are copying it to the Test environment for the first time. Job A references these objects that do not exist in the Test environment:

- Credential Profile for "bobuser"
- Script Library "Parsing Utilities"
- Notification Group "Job Monitors." This group contains notification recipients "tom" and "dave"

The first time you perform the export, all five objects will be listed for import. You complete the import to copy them all to the Test environment.

Next you make a change to "Job A" in the Development environment and need to copy it to Test again. When you do the export and get to the **Select Objects** page, only "Job A" is listed, because the Test server already has the other objects. You complete the import to copy over the changes to the job.

Next you make the following changes:

- In the Test environment, you modify the Notification Group "Job Monitors" because you want it to send notification to different recipients in the Test environment than it does in the Development environment.
- In the Development environment, you modify the Script Library "Parsing Utilities" to make a code change
- In the Development environment, you modify the job "Job A" to make a setting change.

Now you export "Job A" again. Once again the **Select Objects** page only shows "Job A." If you proceed, the changes to "Job A" will get copied over, but the changes to the "Parsing Utilities" library will not. To copy these changes as well, you must uncheck **Don't import referenced objects...**. When you do this, the object selection list will be updated to show all of the data exported from the Development environment.

Now you must be careful to import only the objects that you specifically want, to avoid overwriting any changes that you have made in the Test environment. Begin by clicking **Deselect All** to uncheck all the objects. Then check only the ones you want to copy: Job "Job A" and Script Library "Parsing Utilities."

Complete the import wizard to finish copying the data to the Test environment.

Importing Security Settings

How you handle security during the import process will depend on whether you want to use the same security settings (access security for jobs and other objects) in all environments.

If security settings will be the same in all environments, you should check the **Import security settings for objects** option in the **Import Options** window, and all the permissions for each object will be copied to the target environment.

If you want to have different security settings in different environments (for example, developers have permission in the development environment but not in the production environment), you should **not** check this option.

Instead, configure permissions appropriately on the Job Groups in each environment as discussed in the Initial Configuration topic. Let jobs inherit permissions from their Groups, rather than assign permissions directly to jobs.

Then, when you copy the jobs between environments, they will inherit the security settings in the new environment, without requiring any changes to the permissions on the jobs themselves.

Related Concepts

Managing Multiple Environments 354

Related Topics

Managing Multiple Environments with Separate Instances 354
Configuring Jobs for Multiple Environments 359
Propagating Initial Setup 358
Initial Configuration 356

Auditing and Snapshots

Auditing and Snapshots Overview

adTempus features two levels of change tracking:

- **Auditing.** adTempus creates an audit record when an audited event occurs (for example, an object is created, modified, or deleted). The audit record indicates the type and time of the action and which user performed the action, but does not include details of the change. For example, the audit record will indicate that a job was modified, but does not indicate which settings were changed.
- **Snapshots.** adTempus takes a snapshot of the affected object before the change is applied. The snapshot can later be used to compare the before and after versions of the object, to determine exactly what changed.

Audit records and snapshots appear in the Change Log for each object. You can use the Deleted Objects Window to view and recover deleted objects, if a snapshot was taken at the time of deletion.

Auditing and snapshot settings are configured on the Auditing and Snapshots page of the General Server Options window.

Related Topics

Restoring from a Snapshot 365
Window Reference 384
Create Snapshot Window 459
Auditing and Snapshot Settings 434
Deleted Objects Window 392
Object Comparison Report 426

Snapshot Overview

A **Snapshot** captures all the settings for an object (Job, Credential Profile, etc.) and all the objects it depends on at a particular point in time. This is equivalent to making a backup copy of one or more objects.

Snapshots allow you to compare two versions of an object, restore an object to an earlier state, or recover a deleted object.

adTempus can be configured to create snapshots automatically before objects are modified or deleted, and to create a daily snapshot of all objects. Authorized users can also create on-demand snapshots.

Related Topics

Restoring from a Snapshot 365
Window Reference 384
Create Snapshot Window 459
Auditing and Snapshot Settings 434
Deleted Objects Window 392
Object Comparison Report 426

Restoring from a Snapshot

An object snapshot is simply a data export of the object and all the other objects it is linked to. When you restore an object from a snapshot, you are presented with the same Data Import Wizard as if you were importing the object from an export file. You can restore the object itself and also, if appropriate, restore objects it references to the state they were in at the time of the snapshot.

When you are restoring from a snapshot, be very careful about selecting other objects for import. For example, your snapshot may contain Shared Scripts that the job was using at the time the snapshot was created. If you select one of those scripts to be imported, then the script's current settings and code will be replaced from the snapshot, which will affect any other jobs that are currently using the script. In most cases you would **not** want to import the referenced objects. In that case, the restored job would use the **current** version of each referenced object.

Related Concepts

Snapshot Overview 364
Auditing and Snapshots 364

adTempus Console Reference

Console

The adTempus Console is the primary user interface for adTempus, used to monitor and configure adTempus.

The Console is similar to the Microsoft Management Console and other administrative tools. The Console is divided into two main sections:

- The left section provides a "tree" view of the connected servers and the views available for each.
- Selecting a node (or "folder") in the tree displays the associated "view" on the right side of the Console. Each view is used to work with a different kind of object in adTempus. See the Help Contents for information on each of the views.

Most commands are issued through the pop-up menus associated with the tree nodes and the views. Right-clicking a node, view, or item in a view will display a menu of available commands.

The Console can be used to administer any number of adTempus servers. To establish a new connection, use the **Connect Server** command on the **File** menu.

Within the Console all dates and times are by default show using adjusted for the time zone of the server. For your convenience the clock in the adTempus status bar shows the time on the client and the time on the server. If you prefer, you can have dates and times shown in your local time zone instead using the settings available in the Console Options window.

The active view is automatically refreshed at the interval you specify in the Console Options. You can refresh the Console at any time by pressing **F5** or selecting **Refresh** from the **View** menu.

Throughout the Console, only those objects for which you have View permission are visible, and permissions may restrict your ability to modify these objects.

The icons for some of the nodes in the Console Tree change based on the status of the jobs or other items they contain. For example, the **Failed Jobs** node will display a red error symbol when it contains failed jobs to review.

View Reference

Welcome Page

The Welcome page displays news, announcements, and other information about adTempus. The contents of this page are updated from the Web each time the Console starts.

If you do not want the Welcome page to update from the Web, uncheck the **Show content from the Web** box at the bottom of the page.

You can choose to have adTempus go directly to the Jobs view or the Job Monitor view by changing the Startup View option in the Console Options window.

Server Folders

The **adTempus Servers** node folder contains a separate node for each adTempus server the Console is connected to.

Each server node contains a set of nodes for viewing and working with the jobs on that server.

To connect to a new server, choose **Connect Server...** from the **File** menu, or right-click the **adTempus Servers** folder in the Console Tree and select **Add Server...**.

Job Monitor

Location: **Job Monitor** node in main Console Tree

This feature is new for adTempus version 4.

The **Job Monitor** provides a view of recent, active, and upcoming jobs. The list can be filtered to display only the jobs and timeframe you are interested in.

When you select a job instance in this view, the log messages and step details for the instance are displayed at the bottom of the view.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

If you have jobs that run frequently, they can clutter up the Job Monitor and make it difficult to keep an eye on other jobs that run less frequently. Use a filter (see below) to exclude these jobs. You can either filter the job/group selection to exclude these jobs, or add a tag such as "frequent" to these jobs and then create a filter that excludes jobs tagged with "frequent."

The tool bar at the top of the view provides the following options to customize the appearance of the view:

Show Active

Clicking this button toggles the display of active (currently-executing) job instances.

Show Past

Clicking this button toggles the display of recently-completed job instances. The drop-down list to the right of the button allows you to select the time from which completed instances will be displayed. For example, selecting "12 Hours" will display all instances that have run in the last 12 hours.

Show Next

Clicking this button toggles the display of scheduled jobs The drop-down list to the right of the button allows you to select the time from which scheduled jobs will be displayed. For example, selecting "12 Hours" will display all jobs that are scheduled to run in the next 12 hours.

Note that this view only shows jobs configured with Schedule Triggers. Jobs that are triggered by other events (e.g., File Triggers) do not appear here.

Combined View

Clicking this button toggles the display between a combined view (where past, active, and upcoming jobs all appear in the same list) and a separated view, where past, active, and upcoming jobs appear in three separate lists.

Filter

The drop-down list to the right displays the available filters that you have previously defined.

Select "(none)" to turn off filtering and display all jobs.

Select "Edit..." to edit the current filter or create a new filter. This displays the Job Filter window where you can modify filter settings.

Select "Delete" to delete the current named filter.

The "(unnamed)" filter is a "working" or "scratch" filter. If you modify filter settings and do not provide a name to save your settings under, the "(unnamed)" filter is updated.

Filters that you create are saved in your Console settings file and are only available to you, and only on the computer where they are created.

Related Topics

Failed Jobs....369
Alerts....373

Failed Jobs

The Failed Jobs view is similar to the Execution History query except that it only retrieves instances that have failed (or completed with a warning status) and that have not yet been acknowledged.

You may choose whether to include instances that completed with warnings in this view. When this option is checked, the list will include instances that were abandoned, aborted, or killed.

If instances are found that meet the criteria, adTempus puts a failure (✖) or warning (⚠) icon next to this view in the Console Tree to draw your attention to the view. This view is refreshed regularly as long as the server is connected, even if the view is not active. If no failed instances are present, the view's icon will appear dimmed (○).

When you click the **Acknowledge All** button, adTempus marks all instances as "Acknowledged," and they no longer appear in the Failed Jobs View for you or for any other user. To acknowledge individual instances, select them in the list, then right-click and choose the **Acknowledge Failure** command.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Related Concepts

Job Instance	408

Related Topics

Job Monitor	367
Alerts	373
Troubleshooting Failed Jobs	370
Notes	480

Reference

Instance Details Window	409

Failed Jobs

The Failed Jobs view is similar to the Execution History query except that it only retrieves instances that have failed (or completed with a warning status) and that have not yet been acknowledged.

You may choose whether to include instances that completed with warnings in this view. When this option is checked, the list will include instances that were abandoned, aborted, or killed.

If instances are found that meet the criteria, adTempus puts a failure (❌) or warning (⚠) icon next to this view in the Console Tree to draw your attention to the view. This view is refreshed regularly as long as the server is connected, even if the view is not active. If no failed instances are present, the view's icon will appear dimmed (⊗).

When you click the **Acknowledge All** button, adTempus marks all instances as "Acknowledged," and they no longer appear in the Failed Jobs View for you or for any other user. To acknowledge individual instances, select them in the list, then right-click and choose the **Acknowledge Failure** command.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Related Concepts

Job Instance 408

Related Topics

Job Monitor 367
Alerts 373
Troubleshooting Failed Jobs 370
Notes 480

Reference

Instance Details Window 409

Troubleshooting Failed Jobs

When an adTempus job fails, the instance is listed in the job's history and in the Failed Jobs view with a status of "Failed", and the Job Log shows the message "Job finished with status "Failed."

To determine the reason for the failure, you must examine the Job Log (one of the tab pages at the bottom of the window when you select the failed job in the adTempus Console) for other messages for the failed instance, or examine the step that failed.

There are two primary categories of job failure:

- Some problem prevented adTempus from running the job, or one of its steps (an error reported by adTempus).
- One of the steps returned a result indicating failure (an error reported by the scheduled task).

Error Reported by adTempus

When adTempus encounters a problem that prevents it from running a job or job step, it writes a message to the Job Log describing the problem. These problems are generally easy to identify and resolve based on the error message provided by adTempus. For example, if adTempus cannot find the program that it is supposed to be running.

Errors Reported by Tasks

When a step fails because of an error returned by the program or script being run by adTempus, the Job Log will generally have a warning message similar to this:

```
Step failed because the program run by adTempus returned
an exit code of 1, which indicates failure. Consult the
documentation or output from the program for information
on the meaning of this exit code.
```

You can also find the exit code for a step by displaying the details for the failed job instance and looking at the **Steps** page. The step that failed will have a status of "Failed", and will generally have a **Result** value other than 0. This **Result** is the exit code (or return code) returned by the program that adTempus ran.

The exit code is a numeric code that a program returns to adTempus when it finishes running, which can be used by the program to indicate whether it ran successfully. Most applications use an exit code of 0 to indicate success, and values other than 0 to indicate warning or error conditions.

However, there is no formal standard for exit codes, and no generic list of what exit codes mean. **The meaning of the exit code is determined by whoever wrote the program that is being run**. adTempusdoes not know what, for example, an exit code of "42" from your application means. To find out, you must check the documentation of the program that is being run.

How adTempus Determines "Failure"

By default adTempus treats an exit code of 0 as success, and anything else as failure. That is, if the program that adTempus runs returns an exit code other than 0, adTempus will report the step as "Failed".

This behavior can be changed on the Advanced page of the Program Execution Task Properties window. The **Success Criteria** section lets you define the rules adTempus will use to interpret the exit code.

Diagnosing Failure Exit Codes

If your job fails due to a "failure" exit code you must first determine whether the program did actually fail. For example, some programs may use a non-zero exit code to indicate success. In this case, you need to change the Success Criteria as described above.

Once you have determined that the program is in fact failing you must look to that program for information on what caused the failure.

First, check the documentation for the program in question for information on the meaning of the exit code being reported.

Next, look for error messages being reported by the program:

- Does the program produce a log file containing error messages, or write messages to the computer's Event Log? If so, check these.
- If the program is a "console mode" program that writes output to the console when you run it from a command prompt, capture the program's output (see below) and review it for error messages.

Batch Files

When adTempus runs a batch file, the exit code it receives is the exit code from the last program executed by the batch file. To determine where a failure is occurring in a batch file, capture the output from the batch file (see below) and review it for error messages.

Capturing Console Output

Console-mode programs are programs that are designed to be run from the command prompt or automation tool. Instead of showing a graphical user interface, they write output directly to the console window. If an error occurs in such a program it will often write an error message to the console.

When you run a console-mode program in adTempus you can "capture" the output from the program so that you can review it for error messages. To do so, select the Capture screen output from console-mode program option in the Program Execution Task Properties.

After the job runs, display the instance's details in the job history. The **Captured Files** tab will contain a file named "console output.txt" for each step; this file contains the console output for the program and should be reviewed for error messages.

Related Topics

Failed Jobs 369
Notes 480

Keywords

ADT005252

★Alerts

Alerts are messages from the adTempus server about adTempus operation. Alerts are generally issued when adTempus needs to inform you of a problem that is not related to a specific job or job instance (for example, if there is a problem with the adTempus database, or a job's trigger has failed).

Alerts are listed in the **Alerts** view in the Console.

If unacknowledged alerts are listed, adTempus puts an informational (ℹ), warning (⚠), or error (⊗) icon next to this view the Console Tree to draw your attention to it. If no alerts are displayed, the view's icon will be an empty balloon (🗨).

You may choose whether to include informational alerts in this view. If this option is not checked, only warning and error messages will be displayed.

When you click the **Acknowledge All** button, adTempus marks all Alerts in the view as "Acknowledged," and they no longer appear in the Alerts view for you or for any other user. To acknowledge individual Alerts, select them in the list, then right-click and choose the **Acknowledge** command.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Notification for Alerts

adTempus can be configured to send notification messages (e-mail, text message, etc.) when alerts are logged. Click the **Configure notification for alerts** button to open the Alert Notification Configuration Window, where you can configure notification rules.

Related Topics

Job Monitor.......367
Failed Jobs.......369
Alert Notification Configuration Window.......374

Alerts

Alerts are messages from the adTempus server about adTempus operation. Alerts are generally issued when adTempus needs to inform you of a problem that is not related to a

specific job or job instance (for example, if there is a problem with the adTempus database, or a job's trigger has failed).

Alerts are listed in the **Alerts** view in the Console.

If unacknowledged alerts are listed, adTempus puts an informational (ℹ), warning (⚠), or error (⊗) icon next to this view the Console Tree to draw your attention to it. If no alerts are displayed, the view's icon will be an empty balloon (🗨).

You may choose whether to include informational alerts in this view. If this option is not checked, only warning and error messages will be displayed.

When you click the **Acknowledge All** button, adTempus marks all Alerts in the view as "Acknowledged," and they no longer appear in the Alerts view for you or for any other user. To acknowledge individual Alerts, select them in the list, then right-click and choose the **Acknowledge** command.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Notification for Alerts

adTempus can be configured to send notification messages (e-mail, text message, etc.) when alerts are logged. Click the **Configure notification for alerts** button to open the Alert Notification Configuration Window, where you can configure notification rules.

Related Topics

Job Monitor 367
Failed Jobs 369
Alert Notification Configuration Window 374

Alert Notification Configuration Window

Location: In the Alerts view, click **Configure notification for alerts**

This feature is new for adTempus version 4.

The Alert Notification Configuration window allows you to configure adTempus to send notification messages when new messages are reported in the Alerts view.

You can define any number of rules, each of which uses criteria to select alerts based on their severity, type, or specific message ID.

adTempus creates a default rule named "Notify for database errors" that is configured to select alerts that report critical problems with the adTempus database. Such errors may prevent adTempus from functioning correctly and need to be investigated as soon as possible. Therefore we strongly advise that you configure notification recipients for this rule.

The Alert Notification Rule window allows you to edit the properties for each rule.

Notification for messages related to execution of jobs (such as job failure messages) is configured through Responses that run Notification Actions. See How To Send Notification Messages for Failed Jobs.

Related Concepts

Alerts 373

Reference

Alert Notification Rule 375

Alert Notification Rule

Location: Available from the Alert Notification Configuration Window

An Alert Notification Rule defines the settings for sending a notification message when certain Alerts are reported.

Property Pages

Identification Page

Enter a name and an optional description for this rule.

Selection Rules

Options on this page determine which Alerts this rule will apply to.

Include messages with these severities

To select messages with certain severities, select the appropriate options here. If you do not check any options, the severity will be ignored and the rule will apply to all messages that meet the other criteria on this page.

If you want to send notification for all Error alerts, you can check the Error option and ignore the message category and message ID lists.

Include/exclude message categories

Here you can choose to include or exclude specific categories of message. When you select a category, the message list below will be filtered to show the messages for this category.

For each category you can check Include, Exclude, or neither. Your selections affect alert selection as follows:

- If you do not make any selections in this section (include or exclude), the message category will be ignored, and the rule will apply to all messages that meet the other criteria on this page.
- If you check the Exclude box for any category, all messages in that category will be ignored.
- If you do not check the Include box for any categories, the rule will apply to messages from *all* categories except those that you Exclude.
- If you check the Include box for at least one category, the rule will apply to message from *only* the categories that you have Included.

Include/exclude specific messages

Here you can choose to include or exclude specific messages. When you select a category above, the message list below will be filtered to show the messages for this category. Check the **Show messages for all categories** box to remove this filter and see the complete list of messages.

If you want the rule to apply to all messages in a category you do not need to make any selections here; just Include the appropriate category above. This ensures that any messages added to this category in the future will also be selected.

For each message you can check Include, Exclude, or neither. Your selections affect alert selection as follows:

- If you do not make any selections in this section (include or exclude), the message ID will be ignored, and the rule will apply to all messages that meet the other criteria on this page.
- If you check the Exclude box for any message, that message will be ignored.
- If you do not check the Include box for any messages, the rule will apply to *all* messages that meet the other criteria, except those that you Exclude.
- If you check the Include box for at least one message, the rule will apply *only* to the messages that you have Included.

Notification

Recipients

Select the recipients who should be notified when alerts are logged that meet the selection criteria.

Minimum interval

Use this option to limit how frequently notifications are sent for this rule. Once a notification message is sent for this rule, any subsequent alerts that match the rule will be ignored until the specified number of minutes have passed since the last notification.

Subject and Message

Enter a subject and body for the notification message, or leave blank to use the default system-generated subject and message.

You can use variable tokens to insert values such as the message ID, message, etc. Click the Text Edit button () to open a text editor with a list of the available variables.

Notification Severity

Specify the severity (importance) of this message. Notification Recipients, Addresses, and Groups can all be configured to receive only messages that meet specified severity criteria.

Related Concepts

Alert Notification Configuration Window.. 374

Jobs Folder

The **Jobs** folder is the primary administrative view for adTempus. If you have defined Job Groups, they will appear as folders beneath the Jobs folder.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Clicking the Jobs folder or one of the folders beneath it displays the Jobs View, which is divided into two panes:

- Job List (top pane)
- Job Details (bottom pane)

Job List

The top pane of the view lists all adTempus jobs that you have permission to view. If the job is running the **Status** column indicates this. Otherwise, the **Status** column indicates the result of the last instance that executed. See the Job Status Descriptions topic for a list of possible statuses.

The pop-up menu for this panel allows you to create, modify, delete, hold, duplicate, and execute jobs.

To reduce clutter in the Console and simplify administration, you can organize jobs into groups, which appear as folders in the job list.

When you select a group from the tree at the left of the Console, only the jobs and groups in that group are displayed in the Job List. You can move a group or job to a different group by dragging it to the destination group's folder in the tree.

Items in the Job List may display special icons (▶✖⚠📁📁📁) to draw your attention to their status.

Job Details

When you select a job from the Job List, information about that job is displayed on the five tabs at the bottom of the view.

History

The **History** tab lists information about the most recent 20 instances (executions) of the job. For information about older instances, or to search for instances based on various criteria, use the Execution History Query.

The pop-up menu for this list allows you to display the details for the instance and, for active instances, terminate the instance.

Items in the History List may display special icons to draw your attention to their status.

If an instance completed with a failure or warning status, the instance will be marked with an error (✖) or warning (⚠) icon until the instance is acknowledged by an authorized user.

Job Log

The **Job Log** tab lists messages that have been logged for the job. Only the most recent 20 messages are listed; for information on older messages or to search for messages based on various criteria, use the Message Log Query.

Additional, less important messages may be found in the Job Detail Log.

Agents

The **Agents** tab only appears if the adTempus server to which you are connected is a Master Server. The tab lists each agent that the job targets, and the status of the job on that agent.

The pop-up menu for this list allows you to start the job on a specific agent, without running it on the other agents.

Statistics

The **Statistics** tab provides some basic statistics about the job as a whole (including all instances of the job). Statistics for individual instances and steps are available when you view an instance from the History list.

The **Clear job history** link opens the Clear Job History window, which allows you to purge the history for the job and optionally reset the statistics. History information is also purged automatically based on the job's history retention settings.

Change Log

The **Change Log** tab displays the Change Log for the selected job.

How To...

Create a new Job

1. In the Console Tree, select the Group you want to create the job in.
2. Right-click the group's name and select **New Job...** from the pop-up menu.

   --*or*--

   Right-click In the Job List pane, and select **New Job...** from the pop-up menu.

Move a Job or Group to a different Group

1. If necessary, expand the Console Tree so that the target group is visible in the Console Tree.
2. In the Console Tree, select the group that contains the jobs or groups you want to move.
3. In the Job List Pane, select the jobs or groups you want to move.
4. Drag the selected jobs or groups to the target group.

Create a new Group

1. In the Console Tree, select the Group you want to create the group in.
2. Right-click the group's name and select **New Group...** from the pop-up menu.

   --*or*--

   Right-click In the Job List pane, and select **New Group...** from the pop-up menu.

View or modify the properties of a Group

1. In the Console Tree, select the group whose properties you want to view or modify.
2. Right-click the group's name and select **Edit...**or**View...**from the pop-up menu.

Delete a Group

1. In the Console Tree, select the group in the Job List Pane or in the Console Tree.
2. Right-click the group's name and select **Delete** from the pop-up menu.
3. If the group (or any of the groups below the group) contains jobs, you will be asked whether you want to delete the jobs as well, or just to delete group and move the jobs to the Root group.

Queues Folder

The **Queues** folder contains all Queues that have been defined for the server.

Selecting a Queue displays all jobs that are assigned to that queue. The view is identical to the main Jobs View,except that it does not show any job groups (all jobs assigned to the queue are listed, regardless of what group they are in).

You can move a job to a different queue by dragging it to the destination queue in the tree.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Notification Recipients Folder

The Notification Recipients view lists all Notification Recipients and Notification Groups that you have permission to view. From these views you can add, modify, and delete recipients and groups.

Related Topics

Notification Recipient 303
Notification Group 309

Shared Schedules

The Shared Schedules view lists all of the Shared Schedules that you have permission to view. In this view you can add, modify, and delete shared schedules.

Related Concepts

Shared Schedule 438

Reference

Shared Schedule Properties 439

Holiday Definitions

The Holidays view lists all of the Holiday Sets that you have permission to view. In this view you can add, modify, and delete holiday sets.

You will not be permitted to delete any holiday set that is in use on a job.

Holiday Sets are assigned to jobs through the Schedule Trigger.

Related Concepts

Holiday Set 403

Reference

Holiday Set Properties 404

Scripts

Shared Scripts View

The Shared Scripts view lists all of the Shared Scripts that you have permission to view. In this view you can add, modify, and delete shared scripts.

Shared Scripts are used when you want to be able to use the same script in many places in adTempus, without having to maintain multiple copies of the same code. A Shared Script can be referenced anywhere in adTempus that a script can be used.

ℹ This view only lists scripts that were marked as shared. Scripts that were not shared can only be viewed and modified from the object (job, step, task, action, condition, etc.) with which they are associated.

Script Libraries View

The Script Libraries view lists all of the Script Libraries that you have permission to view. In this view you can add, modify, and delete script libraries.

Remote Agents

This folder only appears if the selected server is configured as a Distributed Scheduling Master computer.

The Remote Agents View lists all Remote Agents that have been defined, and shows the current status for each.

Execution History Query

The **Execution History Query** is used to query the execution history of one or more jobs. Use this query to retrieve job instances (executions) that match your specifications.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

When you select an instance, the bottom panels of the view will show the log messages for the instance and information about the steps in the job.

Double-click an instance to view more information in the Instance Details Window.

The print button (🖨) allows you to print or export the list of instances in the same format as the

Job History Report.

Related Concepts

Job Instance 408

Reference

Instance Details Window 409

Message Log Query

The adTempus Message Log is similar to the Windows Event Log and is used by adTempus to report information to users.

Most messages in the log are tied to a job or job instance. The 20 most recent messages for a job can be viewed on the Job Log tab in the Jobs view. To view additional messages, or search for messages matching specific criteria, use the Message Log query.

Be sure to check the **List non-job messages** box if you wish to see messages that are not tied to a specific job.

Double-click a message to see its details.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Reports

Reports

adTempus includes a report execution and viewing facility, allowing you to produce reports on job settings, history, statistics, etc. Select the Reports node in the main Console tree to view the available reports.

Included Reports

adTempus includes the following reports in the Reports view:

- The Job Inventory Report lists job settings and dependencies.
- The Job Execution Schedule Reports list the scheduled execution times, arranged by date/time or by job.
- The Job Accounting Report provides statistics on executions and CPU time usage.
- The Job History Report shows information similar to that contained in the Execution History Query.
- The Security Configuration Report lists all security logins and groups in adTempus and the permissions assigned to each.

- The Object Change Log Report lists all audited changes recorded in the Change Log for a specified time period.

Additional Reports

In addition to the reports listed in the Reports view, adTempus includes the following reporting capabilities:

- You can generate a report of the listed data from the Failed Jobs, Alerts, Execution History Query, or Message Log Query views, or from the History or Job Log panel for any job.
- You can generate a report showing all settings for top-level objects such as jobs, notification recipients, shared scripts, etc. To create the report, right-click the item in the appropriate list and choose the **Configuration Report** command from the context menu.
- You can create a dependency diagram showing all jobs that a job is linked to. To create the diagram, right-click the job in the main job listing and select the **Diagram Job/Chain** command from the context menu.

Create or Customize Reports

adTempus includes a report designer that you can use to customize the included reports or to create your own reports. See the Report Designer topic for more information.

Additional Reporting Resources

Visit the Reporting Discussion Forum for help with creating or customizing reports, or to exchange reports with other users.

Related Topics

Report Designer.......463

Window Reference

Change Log Window

The **Change Log Window** shows a history of changes and snapshots for an adTempus object.

To reach this window, right-click a top-level object (such as a Shared Script, Notification Recipient, etc.) and choose the **Change Log** command. For Jobs, the Change Log is displayed on the Change Log tab at the bottom of the main Job List View.

The information available in the Change Log for an object will depend on the Audit and Snapshot Settings that have been configured for the object's type.

You can report on changes for all objects in a given time period using the Object Change Log Report.

Audit Records

If auditing has been turned on, you will see an entry recording each modification to the object. Each audit entry contains the name of the user who made the change and any comments entered by the user.

If the record has a snapshot icon (📷), this indicates that adTempus took a snapshot of the object *before* the change was applied. See the next section for information on what you can do with the snapshot.

Snapshots

Snapshots may be created when objects are updated as described in the previous section. They can also be created independent of a change audit. For example, adTempus can be configured to take a daily automatic snapshot of all objects in adTempus, and snapshots can be created on demand in some circumstances.

The following actions are available when you select a snapshot in the Change Log:

View

The **View** command loads the object in a read-only property window so you can view the settings of the object at the time of the snapshot. For example, a Job snapshot is displayed in the same Job Properties window that you use to edit a job.

Compare

The **Compare** commands allow you to compare different versions of the object, producing an object comparison report. The following comparisons are available:

- **Before and after change.** This option is only available for Audit records with associated snapshots, and only when adTempus is configured to always create a snapshot for changes.
- **Snapshot to current version.** adTempus compares the version of the object in the snapshot to the current version.
- **Selected snapshots.** This option is available if you have selected two snapshot records in the list. adTempus compares the object between the two snapshots.

Restore

The **Restore** command allows you to restore the object to its state at the time of the snapshot. See the Restoring from a Snapshot topic for more information.

Related Concepts

Snapshot Overview....................364

Auditing and Snapshots....364

Clear Job History

Clear Job History Window

The Clear Job History window (available through the **Clear job history** link on the **Statistics** page in the Job List) allows you to clear the history (list of previous instances and log messages) for a job immediately, instead of allowing the history to be cleared automatically based on the job's retention settings (specified in the job's properties).

When you click **OK**, the history records will be purged immediately and permanently. If you check the **Also reset statistics for the job** box, adTempus will reset the execution statistics (total executions, total execution time, average execution time, total processor time, and average processor time) to zero. If you do not check this option, the statistics will not be affected.

After the history is cleared, adTempus records an audit message in the job's log, indicating the identity of the user who cleared the history.

History records are not included in object snapshots, so clearing the history cannot be undone.

Console Options Window

Location: **Configuration > Console Options**

The **Console Options Window** allows you to set your preferences for the Console. These settings are saved in your user profile on this computer only, and do not affect other users.

General Options

Console Auto-Refresh Interval

The Console Auto-Refresh Interval specifies the interval (in seconds) at which the Console will retrieve the latest data from the adTempus server. When you are connected to many servers, or to a server with a large number of jobs, you may need to increase the interval (refresh less often) to improve the performance of the Console.

With the auto-refresh interval set to a longer interval, you can always refresh the Console by pressing **F5** in any Console view.

Startup View

Select the view you want adTempus to show when the Console first opens.

- **Welcome Page.** The Console will start on the Welcome Page, which shows adTempus news, announcements, forum postings, Knowledge Base articles, etc.

- **Jobs View for default server connection.** The Console will start on the Jobs View for the server connection that you marked as your default server.
- Job Monitor View for default server connection. The Console will start on the Job Monitor View for the server connection that you marked as your default server.

If your Console is connected to more than one adTempus server, you can designate your default connection using the option in the Server Connection window.

Timestamp Display

If the Console is in a different time zone than the adTempus server, you can choose to have timestamps (such as Next Start time, Last Start time, message log timestamps, etc.) displayed either in the time zone of the server (the default setting) or the time zone of the Console.

Reset all hidden messages

This option turns back on all messages you have previously hidden by checking the "Do not show this message again" box when the message was displayed.

Feature Usage Reporting

Check the **Allow the adTempus Console to collect feature usage information** option to allow the adTempus Console to collect and report information about how you use the Console. This information helps us understand how customers are using the software. The information is collected and reported anonymously. It does not include any information from your adTempus data; it just counts of the number of times you use particular features.

Note: This option controls reporting for the adTempus Console only, and each user's preference is saved separately. Feature usage reporting for the adTempus server is controlled through the Server General Options window.

Related Topics

Server Options 431
Messaging Setup 417
File Servers Window 397
Server Security Settings 437
Linked Servers Window 414

Credential Profiles

Credential Profiles Window

Location: **Configuration > Credential Profiles**

The **Credential Profiles** window allows authorized users to view and manage Credential Profiles.

From this list you can view, modify, delete, and add Credential Profiles.

It is not necessary to come to this window to create new Credential Profiles. Credential Profiles are created automatically as necessary when users enter credentials in adTempus (see the Entering User Credentials topic for more information).

You cannot delete Credential Profiles that are being used in adTempus jobs.

Find/Replace

To find places where a Credential Profile is used and optionally replace the profile with another one, right-click the profile in the list and choose the **Find/Replace References** command to open the Object References window.

Additional Options

Security

Click the **Security** button to open the Credential Profiles Security window, which allows you to specify the default security settings inherited by all Credential Profiles.

Related Concepts

Credential Profiles 91

Related Topics

Entering User Credentials 392

Credential Profiles Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The security settings specified here are inherited by all Credential Profiles and therefore are the default settings for all Credential Profiles.

Automatic Permission Granting

When a user tries to use an existing Credential Profile (e.g., when configuring a job) but does not have "Use Credentials" permission for that profile, she will be prompted to enter the password associated with the profile. If she enters the correct password, she will automatically be granted "Use Credentials" permission for the profile.

Once this permission is assigned, the user can use the profile in the future without being prompted for the password, even if the password is subsequently changed.

Available Permissions

The following permissions apply to Credential Profiles:

Permission	Description
Full Control	Permission to perform all actions on the profile.
Use Credentials	Permission to use the credentials on a job or other object. If a user has Use Credentials permission, she can use the profile without knowing the password for the account.
Modify	Permission to modify the properties of the profile.
Delete	Permission to delete the profile.
Administer security	Permission to change the security settings for the profile.
Change owner	Permission to take ownership of the profile.

Credential Profile Properties

Location: From the Credential Profiles Window, edit a Credential Profile

The **Credential Profile Properties** window allows you to view or modify the properties of a Credential Profile.

Property Pages

Credentials

The **Credentials** page defines the basic properties of the Credential Profile. A Credential Profile generally represents a Windows user account, but may also represent other kinds of account.

Applies To

If the Credential Profile is not a Windows user account, this label indicates what this profile is used for. For example, for database login credentials, **Applies To** indicates what database server this profile is defined for.

Domain

For a Windows user account, enter the domain name, if appropriate. If the account is not a domain account, leave this box empty.

User ID

Enter the user ID for the account.

★**Password**

Enter the password for the account.

Comments

Enter comments or notes about the account.

Password Expiration

If the password for this account is changed in the target system (Windows, database server, etc.), it must be updated in adTempus as well. If the credentials are covered by an expiration policy that requires the password to be changed frequently, you can configure adTempus to issue a reminder alert when the password is nearing expiration.

To use this option, check the **Track password expiration** box and the **Password expires every __ days** box and enter the number of days that a password is valid.

For the initial setup or to resynchronize the expiration date, check the **Set expiration date to** box and enter the date when the current password expires.

Optionally check the **Issue a reminder alert when password is near expiration** to have adTempus log an Alert when the password is nearing expiration.

Once you have configured the password expiration days, adTempus will recalculate the expiration date automatically whenever you change the password in adTempus.

Use system context for certain operations

When this option is checked, adTempus will execute some tasks in the system context (i.e., the Windows identity that the adTempus service is running under) rather than the user account context.

For example, some operations (such as starting or stopping a service) can only be performed by a member of the Administrators group on the computer. Therefore to execute a Service Control Task you would have to run the job under an Administrator account. To avoid using an Administrator account for the job, you can use a standard user account instead, and check this option for the Credential Profile. When adTempus executes the Service Control Task, it will do so under its own security context, which has the necessary permissions.

The following tasks operate in the system context when this option is used:

- Service Control Task
- Computer Shutdown Task

All other tasks performed by the job will operate in the user's security context.

Agent Profiles

This feature is new for adTempus version 4.

The **Agent Profiles** page appears only if the adTempus server is configured as a Distributed Scheduling Master server. This page allows you to define different credentials for use on different Agents.

This can be necessary if your Agents are not in the same security domain as the Master. For example, the Credential Profile is configured to use the Windows logon account `.\bob`, which is local to the Master. When a job that uses this Credential Profile is sent to a Remote Agent, these credentials will not be valid on the Agent. In this scenario, you must specify the credentials that will be used for each Remote Agent. When it sends the job to an Agent, adTempus will send the correct set of credentials for that Agent, instead of the master credentials.

The Agent Profiles page lists all Remote Agents defined for the Master. If an override has been defined for that Agent, "(override set)" will appear after the Agent name. Click **Edit** to create or edit an override for an Agent, or **Delete** to remove an override for an Agent.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

Credential Profiles inherit security settings from the defaults specified in the Credential Profiles window.

Automatic Permission Granting

When a user tries to use an existing Credential Profile (e.g., when configuring a job) but does not have "Use Credentials" permission for that profile, she will be prompted to enter the password associated with the profile. If she enters the correct password, she will automatically be granted "Use Credentials" permission for the profile.

Once this permission is assigned, the user can use the profile in the future without being prompted for the password, even if the password is subsequently changed.

Available Permissions

The following permissions apply to Credential Profiles:

Permission	Description
Full Control	Permission to perform all actions on the profile.
Use Credentials	Permission to use the credentials on a job or other object. If a user has Use Credentials permission, she can use the profile without knowing the password for the account.
Modify	Permission to modify the properties of the profile.
Delete	Permission to delete the profile.
Administer security	Permission to change the security settings for the profile.

Related Concepts

Credential Profiles 91

Related Topics

Entering User Credentials 392

Entering User Credentials

Type in the user ID for the account you want to use. For Windows domain accounts, enter the account name in the following form:

```
domain\username
```

After you type the user name, adTempus will check to see if there is already a Credential Profile for that user ID.

- If no Credential Profile exists, the Credential Profile Properties window will be displayed, allowing you to specify the password and other properties for the Credential Profile. You become the owner of this Credential Profile, and can use the security settings to grant permission for other users to use the Profile without needing to know the password.
- If a Credential Profile exists for the user ID, adTempus checks to see if you already have permission to use it.
  - If you already have permission to use the Credential Profile, you are not prompted to enter the password.
  - If you have not already been granted permission to use the Credential Profile, adTempus will prompt you for the password. If the password you enter matches the password stored for the profile, you will be granted permission to use the profile, and in the future will not have to enter the password.

Related Concepts

Credential Profiles 91

Reference

Credential Profile Properties 389
Credential Profiles 387

Deleted Objects Window

Location: **View > Deleted Objects**

The Deleted Objects window allows you to view and restore objects (Jobs, File Service Providers, etc.) that have been deleted from adTempus.

To be able to restore deleted objects, you must have adTempus configured to create a snapshot on object deletion, using the settings on the Auditing and Snapshots page of the General Server options window.

To view or restore objects, select the type(s) of object and optionally a date range, then click **Find** to list the available objects.

Select an object and click **View** to view the deleted version of the object, or **Restore** to restore the object from the snapshot. See the Restoring from a Snapshot topic for more information.

Related Concepts

Snapshot Overview....364
Auditing and Snapshots....364

Duplicate Job Group Window

This window allows you to duplicate a group and all the jobs and groups it contains. To reach this window, right-click a group in the Console Tree or Job List and select the **Duplicate Group** option from the pop-up menu.

When you duplicate a group, adTempus creates a copy of the group and of all the groups and jobs contained within the group. The new group is created as a sibling of the source group (both have the same parent group).

Any links among duplicated jobs (i.e., Job Conditions, Job Control Actions, or Job Triggers that link two jobs that are both duplicated) are updated to link the duplicated jobs.

You will be able to review the results of the copy operation before you save the new groups and jobs.

Name for New Group

Specify the name for the new group. The name must be unique within the parent group.

Hold all jobs in new group

If this box is checked, all of the newly-created jobs will be held (disabled). If this box is not checked, each job will have the same held status as the original from which it is copied.

Prefix new job names with

Check this box if you want to add a prefix to the name of each duplicated job. If you do not specify a prefix, each job will have the same name as the job from which it is copied. Since the new jobs will be in a different group than the originals, the names will still be considered unique.

Copy

Click the **Copy** button to begin the copy process. When you copy a group, all the groups and jobs it contains are copied as well, creating a copy of the group's hierarchy.

Once the copy is complete, click**OK**to save the new groups and jobs, or**Cancel**to discard them.

Execute Job Window

Location: Right-click a job and select the **Run...** command

The **Run** command is used to submit a job to run immediately. This feature can be used to test the setup of a job, or to run an "on-demand" job.

The **Execute Job** window allows you to specify how the job will be run.

General Options

The **General Options** page contains basic execution options for the job

Ignore conditions for the job

If this option is checked, adTempus will ignore any conditions that are defined for the job (this forces the job to execute even if the conditions are not met). Conditions at the step level are not ignored (see next option).

Ignore conditions for individual steps

If this option is checked, adTempus will ignore any conditions that are defined for steps of the job (this forces the step(s) to execute even if the conditions are not met).

Force a new instance of the job if necessary

This option overrides the Multiple Instances option for the job, forcing adTempus to start a new instance of the job even if another instance is already running.

Override queue limits and force the job to run immediately

If this option is checked, adTempus runs the job immediately, even if the Queue to which it belongs is currently at its quota for the maximum number of concurrent jobs. That is, this option will force the Queue to exceed its limit if necessary.

Run only on Master

If this option is checked, adTempus runs the job only on the Master computer, even if the job is associated with Agents.

To run the job only on a specific agent, select the agent from the Agents list and use the **Run**command on the pop-up menu.

Force job to run on Master

If this option is checked, adTempus runs the job on the Master, even if the Queue's Run on Master option is not checked.

Save the selected options as the default options

When this box is checked, adTempus remembers your choices for all of the above settings (settings in the "Options" group) and makes them the default the next time you run a job manually.

Make the job visible on my desktop

This option is available only if you are running the Console on the same computer where the adTempus service is running. When this box is checked, it overrides the job's User Interaction settings and causes adTempus to run the job in your current Windows logon session, if possible. This ensures that you will be able to see any windows shown by programs that the job runs.

This option may not be available if the adTempus administrator has disabled it or depending on other adTempus settings.

Responses

Choose which Responses you want to execute (this option applies to Responses for the job and for steps within the job):

- **Execute Responses:** All Responses are executed as normal.
- **Do not execute any Responses:** No Responses will be executed.
- **Do not execute any Job Control Responses:**All Responses will be executed*except*Job Control Responses (Responses that run, terminate, hold, or release a job or job step). Use this option if you do not want adTempus to execute any other jobs that are chained to the current job.

Checkpoint

Specify the checkpoint to pass to the job.

Steps

The options on the **Steps** page determine which steps adTempus will run.

Run the job from the beginning

The job will run as normal, with all steps executed.

⭐Run the job from the specified step

Execution will begin with the selected step and continue to the end of the job. If you check **Run only the selected step**, the job will end after the selected step is run.

Steps that have been disabled are not listed.

Run only the selected steps

Only the steps you select will be executed. Steps will be executed in their original order (as listed here), but you may skip steps.

Steps that have been disabled are not listed.

Variables

The **Variables** page allows you to set or override Job Variables that have been defined for this job. Values you provide here are used only for this execution of the job.

Notes

The **Notes** page displays the extended description/notes saved in the job definition.

You may also specify comments for this instance of the job (for example, an explanation for why the job is being run at an unusual time). Comments are saved with the instance and can be viewed in the instance details.

File Servers

File Service Provider

This feature is new for adTempus version 4.

A **File Service Provider** specifies the settings that adTempus uses to connect to a file server for a File Trigger or File Operation Task.

File Service Providers are managed using the File Servers Window.

adTempus supports the following kinds of file server:

- FTP
- SFTP (SSH)

Related Topics

File Servers Window 397
FTP Service Provider 397
SFTP Service Provider 400

File Servers Window

Location: **Configuration > File Server Setup**

The **File Servers** window allows you to define file servers that can be used by File Triggers and File Operation Tasks to work with files on remote servers.

The following server types are available:

- FTP
- SFTP (SSH)

Security

The **Security** button opens the Filer Server Security Window, where you can edit the default permissions that apply to all File Servers and specify which users are allowed to create new servers. Security for an individual server can be edited in the property window for the server.

Related Concepts

File Servers 396

Related Topics

Server Options 431
Messaging Setup 417
Server Security Settings 437
Linked Servers Window 414
Console Options Window 386

FTP Service Provider

Location: Available from the File Servers Window

The **FTP Service Provider Properties** window contains the settings used to connect to an FTP server for File Triggers and File Operation Tasks.

Property Pages

FTP Settings

Server Type

Select the type of security used by the FTP server.

If you are connecting to an SFTP server, create an SFTP Service Provider instead. SFTP is not the same as FTP.

Server Address

Enter the DNS name or address of the FTP server.

Port

Enter the port to connect to. For standard (unsecured) FTP and FTP with Explicit SSL this is generally port 21. For FTP with Implicit SSL this is generally port 990.

Authentication

Enter the user ID and password for connecting to this server.

Use unencrypted control channel

If this option is checked, the FTP control channel will be unencrypted even if the data channel is secured. Check this option only if required for your FTP server.

Validate server certificate

Uncheck this option to bypass validation of the server's SSL certificate.

Use Mode Z to compress data

If supported by the FTP server, check this option to compress data during transfer.

Use Passive mode

Used Passive mode for transfers. This may be required for some FTP servers and/or network configurations.

Preserve timestamp on upload

When this option is checked, adTempus will attempt to preserve each file's timestamp when uploading files. If the option is not checked, the server will set the timestamp to the current date/time. Not all FTP servers support setting the timestamp.

Preserve timestamp on download

When this option is checked, adTempus will attempt to preserve each file's timestamp when downloading files. If the option is not checked, adTempus will set the timestamp to the current date/time.

Server time zone

Specify the time zone in which the FTP server is located. This information is needed for adTempus to accurately preserve file timestamps on download.

Proxy

If adTempus needs to go through a proxy server to connect to the file server, specify the proxy information here.

Proxy Type

Select the type of proxy server:

- None (no proxy server is used)
- HTTP
- SOCKS4
- SOCKS5
- FTP

Server Address

Enter the DNS name or address of the proxy server.

Port

Enter the port to connect to on the proxy server.

Authentication Type

Select the type of authentication that the proxy server uses, and enter the user ID and password if needed.

The supported authentication types for HTTP proxies are:

- Basic
- NTLM

The supported authentication types for FTP proxies are:

- Automatic. adTempus will attempt to determine the correct authentication type
- SITE host
- USER user@host
- USER with login
- USER/PASS/ACCT
- OPEN host

Security

The Security page contains a standard Security Editor used to manage permissions for this server definition.

Test

When you click the **Test** button adTempus will attempt to connect to the FTP server using the settings you have provided.

Related Concepts

File Servers 396

SFTP Service Provider

Location: Available from the File Servers Window

The **SFTP Service Provider Properties** window contains the settings used to connect to an SFTP (SSH) server for File Triggers and File Operation Tasks.

If you are connecting to an FTP server, create an FTP Service Provider instead. SFTP is not the same as FTP.

Property Pages

SFTP Settings

Server Address

Enter the DNS name or address of the SFTP server.

Port

Enter the port to connect to. This is generally port 22.

Authentication

The following forms of authentincation are supported for SFTP. Select the option required by your server.

- **User ID and Password.** Enter the user ID and password for the account.
- **User ID and Private Key.** Enter the user ID for the account.

  Then select the file that contains your private key (provided by the server administrator). adTempus supports private keys in OpenSSH format (file extension .pem) or PuTTY format (file extension .ppk). Once you have selected the file, the contents are saved in adTempus and the original file is no longer needed by adTempus.

  You also must enter the private key password (the password used to protect the private key).

- **User ID, Password, and Private Key.** Enter all user ID, password, and private key information as described for the two options above.

Preserve timestamp on upload

When this option is checked, adTempus will attempt to preserve each file's timestamp when uploading files. If the option is not checked, the server will set the timestamp to the current date/time.

Preserve timestamp on download

When this option is checked, adTempus will attempt to preserve each file's timestamp when downloading files. If the option is not checked, adTempus will set the timestamp to the current date/time.

Proxy

If adTempus needs to go through a proxy server to connect to the file server, specify the proxy information here.

Proxy Type

Select the type of proxy server:

- None (no proxy server is used)
- HTTP
- SOCKS4
- SOCKS5

Server Address

Enter the DNS name or address of the proxy server.

Port

Enter the port to connect to on the proxy server.

Authentication Type

Select the type of authentication that the proxy server uses, and enter the user ID and password if needed.

The supported authentication types are for HTTP proxies are:

- Basic
- NTLM

Security

The Security page contains a standard Security Editor used to managed permissions for this server definition.

Test

When you click the **Test** button adTempus will attempt to connect to the SFTP server using the settings you have provided.

Related Concepts

File Servers 396

Holiday Set

A holiday set is a special kind of shared schedule that can be used to specify days on which a job should *not* be run (using the settings on the Holidays page of the Schedule Trigger window).

To view or manage holiday sets, select the Holiday Definitions folder in the adTempus Console.

adTempus comes preconfigured with the "Standard U.S. Holidays" set, which includes standard U.S. holidays:

- New Year's Day (January 1)
- Martin Luther King Day (Third Monday in January)
- Presidents' Day (Third Monday in February)
- Easter
- Memorial Day (Last Monday in May)
- Independence Day (July 4)
- Labor Day (First Monday in September)
- Columbus Day (Second Monday in October)
- Veterans' Day (November 11)
- Thanksgiving (Fourth Thursday in November)
- Day After Thanksgiving
- Christmas Day (December 25)

This set can be altered to match the holiday schedule of your organization, or you may create new holiday sets.

Related Topics

Schedule Trigger Properties 243

Holiday Definitions 381

Reference

Holiday Set Properties 404

Holiday Set

A holiday set is a special kind of shared schedule that can be used to specify days on which a job should *not* be run (using the settings on the Holidays page of the Schedule Trigger window).

To view or manage holiday sets, select the Holiday Definitions folder in the adTempus Console.

adTempus comes preconfigured with the "Standard U.S. Holidays" set, which includes standard U.S. holidays:

- New Year's Day (January 1)
- Martin Luther King Day (Third Monday in January)
- Presidents' Day (Third Monday in February)
- Easter
- Memorial Day (Last Monday in May)
- Independence Day (July 4)
- Labor Day (First Monday in September)
- Columbus Day (Second Monday in October)
- Veterans' Day (November 11)
- Thanksgiving (Fourth Thursday in November)
- Day After Thanksgiving
- Christmas Day (December 25)

This set can be altered to match the holiday schedule of your organization, or you may create new holiday sets.

Related Topics

Schedule Trigger Properties 243
Holiday Definitions 381

Reference

Holiday Set Properties

The **Holiday Set Properties** window allows you to view or edit a Holiday Set.

Property Pages

Schedule

The **Schedule** page defines the days that constitute holidays.

Name

Provide a descriptive name for this holiday set. The name must be unique across all holiday sets and shared schedules.

Enable this schedule

Check the box to enable the holiday set or clear the box to disable it. If a holiday set is disabled, it will not appear in the list of available holiday sets when you are creating a job. Also, it will no longer cause jobs that use it to not execute on the holidays.

Rules for all years

Specify any number of specific dates or date rules to define the days that are holidays. See the Select Days topic for information on specifying days. The rules you enter here will apply to all years.

Rules for specific years

If you have rules that only apply to specific years, enter them here. For each rule you may choose whether to apply the rule to all years, to a single year, or to a range of dates.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Holiday Sets:

Permission	Description
Full Control	Permission to perform all actions on the holiday set.
List/User	Permission to use the holiday set.
View	Permission to view the properties of the holiday set.
Modify	Permission to modify the properties of the holiday set.
Delete	Permission to delete the holiday set.
Administer security	Permission to change the security settings for the holiday set.
Change owner	Permission to take ownership of the holiday set.

Permission to create new Holiday Sets is controlled through the Shared Schedule and Holiday Security window.

Related Concepts

Holiday Set 403

Related Topics

Schedule Trigger Properties 243
Holiday Definitions 381

Job Filter Window

Location: Available from the Job Monitor View

The **Job Filter** window allows you to filter the jobs that are displayed in the Job Monitor.

Select Jobs and Groups

By default all jobs are shown. Use the selection boxes to display only certain jobs, or only jobs from certain Groups.

Tags

Enter tags to filter the job selections. For example, if you enter `prodA prodB`, only jobs tagged with `prodA` *or* `prodB` will be displayed. To exclude jobs with a certain tag, prefix the tag with minus ("-"). For example, if you enter `prodA prodB -frequent`, only jobs tagged with `prodA` *or* "`prodB` but *not* `frequent` will be tagged.

Save settings as

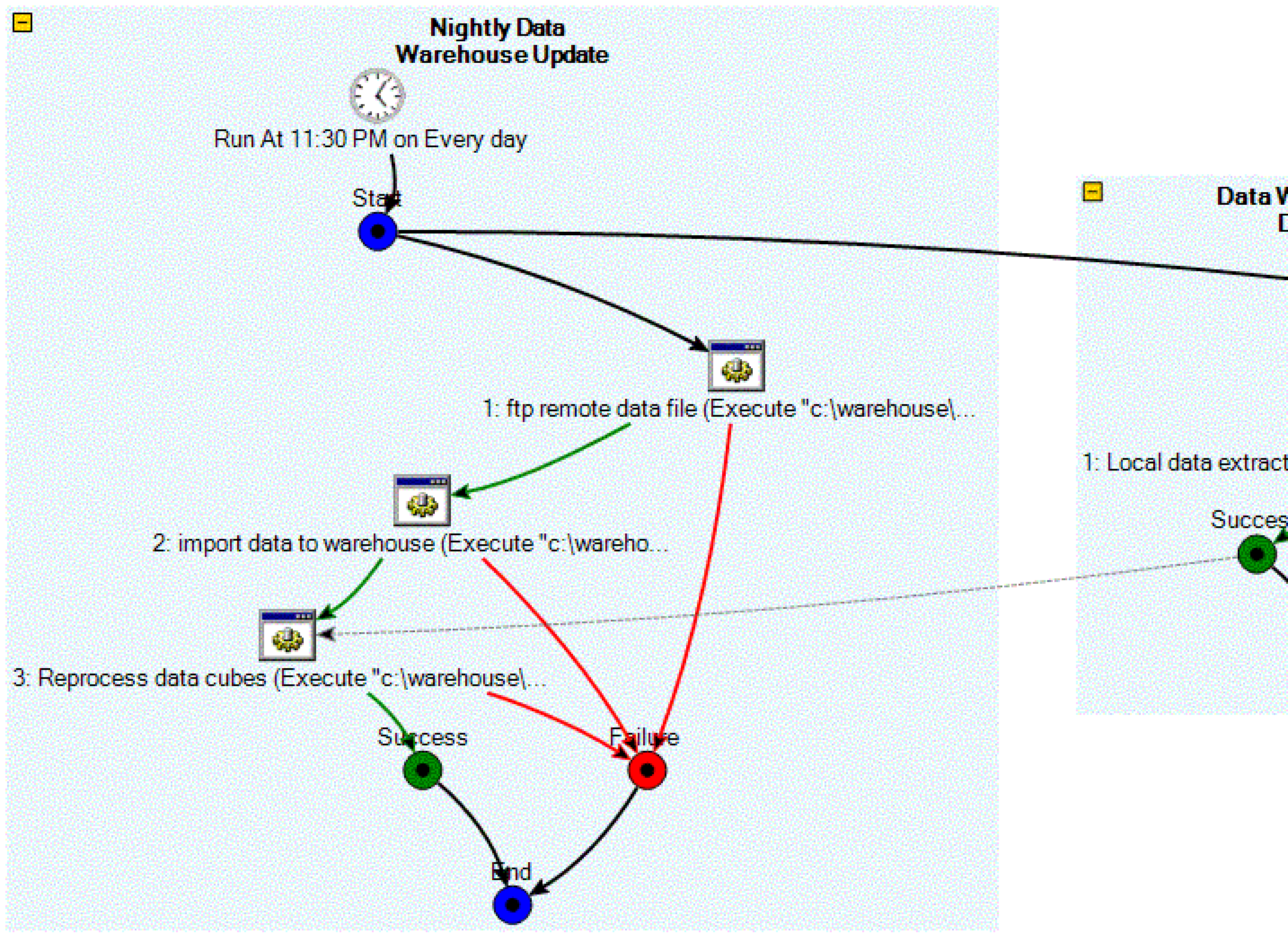
To save these filter settings for use in the future, enter a name for the settings. If you leave the name empty, the settings will update the `(unnamed)` filter.

Job Flow Diagram

The **Job Flow Diagram** tool allows you to create job flow diagrams, which are graphical representations of your jobs that can be used to review and document your job setup.

To diagram a job chain, right-click a job in the Job List and select the **Diagram job/chain...**command. adTempus will diagram the job and the complete chain of all jobs linked to it (and jobs linked to those jobs, and so on).

Diagrams are created automatically based on your job setup. For example, this diagram shows the "Nightly Data Warehouse Update" job and the "Data Warehouse Local Data Extract" job from the job setup example.


⊟
Nightly Data
Warehouse Update
Run At 11:30 PM on Every day
Start
1: ftp remote data file (Execute "c:\warehouse\...
2: import data to warehouse (Execute "c:\wareho...
3: Reprocess data cubes (Execute "c:\warehouse\...
Success
Failure
End
⊟
Data W
D
1: Local data extract
Succes

The diagramming tool does not allow you to build or edit your job graphically. You can display and edit the properties for jobs, steps, actions, and links by double-clicking them in the diagram.

Navigating the Diagram

Each job is represented by a shaded box. Initially the diagram will show the complete internal details of all of the selected jobs (and the jobs linked to them). Each trigger, condition, step, and action for the job is displayed. Links between jobs are also shown.

To hide or show the details for an individual job, use the collapse (⊟) and expand (⊞) buttons found on each job. You can also expand (⁺₊) and collapse (⁻₋) all jobs at once.

By default, jobs, triggers, steps, and actions are labeled, but links are not. To see a description of a link, click it, and information will be displayed in the Properties box to the right of the diagram. You can turn on display of labels if you prefer, using labels toggle button (💬).

Link Types

Links are color-coded to allow you to get a sense of the job flow without needing labels on the links:

Solid black link →	Represents a course that is always taken
Green link →	Represents a course that is taken on success
Red link →	Represents a course that is taken on failure
Yellow link →	Represents a course that is taken in "warning" conditions
Grey dashed link ⇢	Represents a condition that must be met (does not indicate a direct action). In the sample diagram above, the end node of "Data Warehouse Local Data Extract" has a condition link to step 3 of "Nightly Data Warehouse Update." This indicates that step 3 of "Nightly Data Warehouse Update" cannot run until "Data Warehouse Local Data Extract" completes, but does not mean that "Data Warehouse Local Data Extract" directly runs step 3 of "Nightly Data Warehouse Update."

Editing Objects

You can edit an object (job, step, link, action, condition, or trigger) by double-clicking it on the diagram. If you make any changes, the diagram will be redrawn automatically.

Printing the Diagram

The diagram can be printed using the Print (🖶) and Print Preview (🔍) buttons on the Toolbar.

Saving the Diagram

Diagrams can be saved to a file using most common graphics image formats.

Job Instance

Each time a job is executed, this is called an "instance" of the job. Information about each instance is saved in adTempus, based on the job's history retention settings.

You can review the history for a job by selecting the job in the Console's Job List View or using the Execution History Query.

Instance Acknowledgment

When a job instance ends with an error or warning status, it is flagged for review and appears in the Failed Jobs view in the Console. An authorized user can "acknowledge" the failure or warning, clearing the flag status of the instance and removing it from the view.

Related Topics

Execution History Query 382
Failed Jobs 369

Reference

Instance Details Window 409

Job Instance

Each time a job is executed, this is called an "instance" of the job. Information about each instance is saved in adTempus, based on the job's history retention settings.

You can review the history for a job by selecting the job in the Console's Job List View or using the Execution History Query.

Instance Acknowledgment

When a job instance ends with an error or warning status, it is flagged for review and appears in the Failed Jobs view in the Console. An authorized user can "acknowledge" the failure or warning, clearing the flag status of the instance and removing it from the view.

Related Topics

Execution History Query 382
Failed Jobs 369

Reference

Instance Details Window 409

Instance Details Window

The Job Instance Details window shows the details for a single job instance (a single execution of a job).

Detail Pages

Job

The **Job** page lists basic information about the instance.

Instance

The instance ID for this instance. Instances are numbered sequentially.

Status

The status of the instance.

Last Step

The last step executed in the instance.

Last Checkpoint

The last checkpoint reported for the job.

Execution Start

The date/time at which execution started (dates and times are shown using the time zone of the server).

Execution Finish

The date/time at which execution finished (dates and times are shown using the time zone of the server).

Elapsed Time

The total elapsed time for the job (the difference between the Start and Finish times).

Processor Time

The amount of CPU time used by the job. Note that processor times listed for jobs and steps only include program execution tasks and script tasks. Other tasks do not update the processor times.

Execution Reason

The reason the job was executed.

Cycle ID

The Cycle ID for this job, if one was assigned.

Chain ID

The Chain ID links together all instances that are part of a chain of jobs linked together by Responses and Job Triggers. For example, if an instance of Job A runs a Response when it finishes that starts an instance of Job B, both instances will have the same Chain ID. This value is intended for use when reporting or querying adTempus job history.

Log

The **Log** page lists messages logged by adTempus related to this job. Messages for the job and for individual steps are listed. To view the details of a message, double-click the appropriate row in the list.

Additional low-level messages intended for diagnostic/debugging purposes may be reported in the Job Detail Log and/or Script Debug Log, which can be found on the **Captured Files** page.

Conditions

The **Conditions** page lists information about the status of each Condition for the job or its steps. For example, if a job is in Waiting for Conditions status, this page will show you which condition(s) are still unmet.

Steps

The **Steps** page lists information for each step of the job. For more information on a step, double-click the appropriate row to display the Step Details.

Only steps that were executed (or that adTempus attempted to execute) are listed. If the step was never reached (e.g., because the job was aborted or failed before reaching the step, or because a job control action within the job caused the step to be skipped over), the step will not be listed here.

You can change the way this grid (listing) appears as follows:

- Click on a column heading to change the sort order.
- Click and drag a column heading to change the order of columns.
- Right-click any column heading to display a list of available columns, allowing you to select which columns appear in the grid.

Changes are saved in your local user profile and will be used again the next time you run the adTempus Console.

Captured Files

The **Captured Files** page lists any files captured by the job (using File Capture actions or the Capture Console option).

Captured console output has the name `console output.txt`.

Some additional special files may be present:

- ⭐ The Job Detail Log contains details of job and step execution, which can be used for debugging purposes.
- Script Debug Logs contain trace messages written by user scripts.

To view or save a captured file, right-click the appropriate row for a menu of options.

- Select **Open** to open the file using the default application for the file type.
- Select **Save As...**to save the file to the location of your choice. This saves a copy of the file that you can do with as you wish; the captured file is still saved on the server as part of the job history.

Comments

The **Comments**page allows authorized users to record comments about this job instance. For example, if the job failed, the system operator can record notes about what error resolution procedures were followed.

Related Concepts

Job Instance 408

Related Topics

Execution History Query 382
Failed Jobs 369

Job Detail Log

Location: Available from the Captured Files page of the Job Instance Details window

This feature is new for adTempus version 4.

The Job Detail Log contains information generated by adTempus during execution of the job that may be useful in diagnosing problems with job execution or unexpected results.

It contains information about the values of all Job Variables at the time of execution plus other details specific to particular tasks.

The Job Detail Log appears as a Captured File attached to each instance of a job.

Step Details

The **Step Details** window (reached from the Steps page of the Job Instance Details window) lists information about a job step executed in a job instance.

Step

The step number of the step. Steps are numbered sequentially, beginning with 1. The step number reflects the sequence of steps at the time the job was executed.

Description

The step's description, at the time the job was executed.

Status

The status of the step.

Result

The result returned by the Task. For a Program Execution Task, this will be the exit code (return code) returned by the process. For Script Execution Tasks, this will be the return value from the script. For other task types, no result value is defined.

Execution Start

The date/time at which execution started (dates and times are shown using the time zone of the server).

Execution Finish

The date/time at which execution finished (dates and times are shown using the time zone of the server).

Elapsed Time

The total elapsed time for the step (the difference between the Start and Finish times).

Processor Time

The amount of CPU time used by the step. The processor time is only valid for Program Execution tasks. Other tasks do not update the processor times.

Process ID

Applies to Program Execution tasks only. Indicates the Windows Process ID for the process executed by the step.

Last Checkpoint

The last checkpoint reported for the step.

Job Variable Properties

The **Job Variable Properties** window allows you to view or modify the definition for a Job Variable.

Properties

Variable Name

Enter a name for this Variable. The name must be unique within the list of variables it is being added to. If the object you are adding the Variable to already has a variable with the same

★name, you cannot change the name or add a new Variable with the same name. However, you can edit the existing variable to override its value.

Variable names can be up to 255 characters in length and are not case sensitive ("MyVariable" is the same as "myvariable").

Include in program's environment variables

If this option is checked, this variable will be included in the environment variables for any programs run by job steps that use these variables. If the option is not checked, the variable will be available for use within adTempus but will not be in the program's environment variables.

Hide Value

If this option is checked, the value of the variable will not be visible in lists in adTempus. The value will also not be visible in this window unless you unhide it or click the **Reveal value** button. Only users with permission to modify the variable can reveal the value.

Variable Type

Select the type of data this variable holds. If the variable will hold a date, time, or numeric value, setting the type appropriately allows you to format the value when inserting it.

Value

Enter the value for this Variable. There is no limit on the length or content of the value for string variables.

Description

Optionally, enter a description or notes about this Variable.

Related Concepts

Job Variables 88

Related Topics

How To: Set and Retrieve Variables in Scripts 500
Job Variable Condition 269
Job Variable Update Action 289
Job Variable Update Task 178
Predefined Variables 485
Text Edit Window 441

Linked Servers

Linked Server

Job Control Actions and Job Conditions allow jobs on one adTempus server to execute or have dependencies on jobs on another adTempus server. Before this can be done, the two adTempus servers must be linked together to establish communications and a trust relationship between them.

Linked Servers are distinct from Remote Agents. It is not necessary for a computer to be defined as a Linked Server in order for it to be a Remote Agent.

Use the Linked Servers window to manage Linked Servers. You must be a member of the adTempus Administrators group on both adTempus servers in order to create or modify a link between them.

Related Topics

Linked Servers Window........414

Reference

Linked Server Properties........415

Linked Servers Window

Location: **Configuration > Linked Servers**

The **Linked Servers** window shows a list of all adTempus servers that the current adTempus server has a trust relationship with. You must be a member of the adTempus Administrators group to create, modify, or delete trust relationships.

Click **Add** or **Edit** to display the Linked Server Properties window to create or modify a trust relationship.

Click **Delete** to remove a trust relationship. If jobs are still configured with Job Control Actions or Job Conditions that link to this server, those actions or conditions will fail at execution.

To add a new trust relationship, the adTempus Console must be connected to both servers, and you must be a member of the adTempus Administrators group on both adTempus servers.

Upgrading from Previous Versions

If you are upgrading from adTempus 3 or earlier and you have Job Conditions or Job Control Actions that target remote computers, you will need to create server links for those remote computers. In this case adTempus will log an Alert with ID 5270 at service startup, indicating which computers need to have trust relationships added.

To establish the relationships:

1. Make sure the Console is connected to the source server (the server where the conditions or actions are configured) and to any remote servers listed in the Alert. (You must be a member of the adTempus Administrators group on all affected servers.)
2. Select the source server's node in the Console Tree and open the **Linked Servers Window**.
3. For each remote computer, use the **Add** button in this window to add a linked server. Be sure to check **The current server can send commands** option in the Linked Server Properties.

Related Concepts

Linked Servers 414

Related Topics

Server Options 431
Messaging Setup 417
File Servers Window 397
Server Security Settings 437
Console Options Window 386

Reference

Linked Server Properties 415

Linked Server Properties

Location: In the Linked Servers Window, create or edit a Linked Server

A Linked Server defines a connection and a trust relationship between two adTempus servers that are not in a Master/Agent relationship.

Server to Add

When you are adding a new Linked Server, select the server you want to add the link with. Before you can add a link to a server, you must be connected to it in the adTempus Console.

When you add a Linked Server definition, the current server will appear as a Linked Server on the target server.

The current server can send commands...

Check this option to allow the current server to send commands (such as job execution commands) to the target server.

The target server can send commands...

Check this option to allow the target server to send commands to the current server.

The target computer is not in the same domain as the current computer

If the two adTempus servers are not in the same Windows domain (and do not have a trust relationship defined at the Windows level), you must install server certificates for both servers and check this box. See the Server Certificates topic for more information.

Override the server address/connection information

If the adTempus server needs to use a different name or address than the Console is using to connect to this server, enter the information here.

Related Concepts

Linked Servers 414

Related Topics

Linked Servers Window 414

Network Resource

Network Resource

The Network Resource allows you to specify a network connection that must be established in order for a job to run. Using the network resource you may optionally map network drive letters for the job.

For more information on network access for jobs, see the Network Access topic.

Network resources for a job are managed from the Resources page of the job's properties.

Related Concepts

Network Access for Jobs 482
Resources 90

Reference

Job Resources Page 141
Network Resource Properties 416

Network Resource Properties

The **Network Resource Properties** contains settings for a network connection that needs to be made for your job.

See the Network Access topic for important information on network resources. When possible you should avoid using mapped drive letters, and use UNC paths instead.

Generally there is no need to explicitly connect network resources unless you need to assign a drive letter or specify user credentials other than those used by the job. If neither of these is the case, your tasks can refer to the resources by their UNC paths without explicitly making a connection.

Network path to connect

Specify the network path to connect to, using UNC notation. For example, specify `\\prod1\data` to connect to the `data` share on the `prod1` server.

Assign drive letter

Check this option to assign a drive letter to the path you have selected. The drive letter will be mapped to the network connection for the duration of the job, allowing your tasks to refer to the resources using drive letters.

Drive letter mappings may fail for a variety of reasons. Therefore you should avoid using drive letters if possible. See the Network Access topic for more information.

User Account

Specify the credentials adTempus should use when making the network connection. If the account that the job is running under has access to the resource, there is no need to specify credential here: specify credentials only if you need to user a User ID different from the one the job is using.

Related Concepts

Network Resource..416

Messaging Setup

Messaging Setup Window

Location: **Configuration > Notification and E-Mail Account Setup**

The Notification Setup Window contains the server settings used for sending and receiving e-mail messages and for sending SMS (text) and XMPP (Jabber) notification messages. These are referred to as Messaging Service Providers.

SMTP Servers

The SMTP servers are used to send e-mail messages. You must define at least one SMTP server in order for adTempus to be able to send e-mail messages from Notification Actions or Notification Tasks.

For redundancy you can define more than one SMTP server. When sending a message adTempus will try the servers in the order they are listed, until it is able to deliver the message successfully.

See Also: SMTP Server Properties

Incoming E-Mail Servers

Incoming e-mail servers allow adTempus to connect to e-mail inboxes for E-Mail Triggers and E-Mail Processing Tasks. Each Provider represents a connection to a POP3 or IMAP server.

See Also: Incoming E-Mail Provider Properties

SMS Gateways

SMS gateways are used by adTempus to send SMS messages (text messages). Define one or more SMS service provider here. When you create a Notification Address that uses SMS notification, you may optionally link the address to a specific provider. If the address is configured to Use any available connection, adTempus will try the SMS gateways in the order they are listed here.

See Also: SMS Service Provider Properties

Jabber (XMPP) Servers

Jabber (XMPP) servers are used to send instant messaging notifications using the Jabber (or XMPP) protocol, which is used by many services including Google Talk.

See Also: Jabber Service Provider Properties

Additional Options

Security

The **Security** button opens the Messaging Service Provider Security Window, where you can edit the default permissions that apply to all Messaging Service Providers and specify which users are allowed to create new providers. Security for an individual provider can be edited in the property window for the provider.

Related Topics

Server Options 431
File Servers Window 397
Server Security Settings 437
Linked Servers Window 414
Console Options Window 386
Notification Task 181
Notification Action 291
Notification Recipients 93
Notification Recipient 303
Notification Group 309
How To: Send Notification Messages for Failed Jobs 493

SMTP Server Properties

Location: Available from the Messaging Setup Window

An SMTP server is a server used to send e-mail messages. Your mail server may be internal to your organization or it may be hosted elsewhere (such as a Gmail server).

Description

Enter a name or brief description for this server, such as "Primary mail server" or "Corporate mail server."

Enabled

Clear this box to disable the server. adTempus will not send messages through this server while it is disabled.

Server Name or Address

Enter the DNS name or IP address of the server. If the mail server runs on the same computer as adTempus, use `127.0.0.1` as the address.

Port

Enter the port that the SMTP server is listening on. Most servers use the default port, which is 25. Leave this box empty to use the default, or specify a value if you server uses a different port.

If you are using a mail server such as Gmail that requires a secure connection, you must specify port 465 (SSL) or 587 (for TLS).

Return Address for Messages

Specify the return address to use on e-mail messages sent to the server. The address **must be specified**, and must be a valid address. Your mail server will reject messages that do not have a valid return address.

Sender Display Name

The display name to use in the return address. By default messages will use "adTempus on *servername*" (where *servername* is the name and instance name of the adTempus server) as the name of the sender. If you want to use a different name, specify it here. If you include the token `%servername%`, adTempus will replace that with the name of the adTempus server (and the instance name, if other than the default).

You can specify different return addresses for different jobs in any of the following ways:

- Set the **E-Mail Sender** options in a Notification Task or Notification Action.
- Set the **NotificationFromName** and/or **NotificationFromAddress** Job Variables for a job, group, queue, step, etc. You can do this either through the properties window for the job, group, etc., or by running a script within the job.

Maximum Size for Messages

Set the maximum message size allowed by the mail server. If a notification message exceeds this size due to the length of the message text, adTempus will truncate the message text. If message attachments cause the message to exceed this length, adTempus will selectively exclude attachments from the message to keep it within the size limits (a notice will be appended to the message indicating which attachment(s) have been dropped).

If you are sending notification messages that include attachments, it is important to set this value correctly to ensure that notification messages are not rejected by your mail server for being too large.

Authentication

If your mail server is an internal server it probably does not require authentication. If it is external (Gmail and other hosted services), it almost certainly does.

If your server requires authentication, enter the necessary information here. Check with your mail system administrator if you are unsure regarding these settings.

Generally the **User ID** will be your complete e-mail address (e.g., "my.name@example.com").

Specifying authentication information for a mail server that does not accept it will cause messages to fail.

Test...

Click the **Test...** button to send a test message using the settings you have provided. If adTempus is unable to send the test message it will display an error message with the reason.

Related Topics

Messaging Setup 417

Incoming E-Mail Provider Properties

Location: Available from the Messaging Setup Window

An **Incoming E-Mail Provider** allows adTempus to connect to e-mail inboxes for E-Mail Triggers and E-Mail Processing Tasks. Each Provider represents a connection to a POP3 or IMAP server, using a particular set of login credentials.

If you need to monitor more than one mailbox (i.e., log on using more than one user account), you must create a separate Provider for each account.

Property Pages

Provider

Provider Name

Enter a unique, descriptive name for this provider. For example, "Automation Trigger IMAP account."

Server Name or Address

Enter the name or IP address for the mail server.

Server Port

Enter the port number, or leave blank to use the default port for the server type.

If you are using a mail server such as Gmail that requires a secure connection, you must specify the correct port number:

- Secure POP3 (SSL-POP): 995
- Secure IMAP (IMAP4-SSL): 585
- IMAP4 over SSL (IMAPS): 993

Server Type

Select the type of server: POP3 or IMAP.

User ID and Password

Enter the credentials used to connect to this server.

Test

Click the **Test...** button to verify that adTempus can connect to the server using the settings you have provided. If adTempus is unable to connect it will display an error message with the reason.

Security

The Security page contains a standard Security Editor that allows authorized users to manage the security for this Provider. Permission to create new providers is controlled through the Notification Recipient Security options.

Related Topics

Messaging Setup..417

SMS Service Provider

The SMS Network Service Provider supports text message notification using the Short Message Service (SMS) protocol. Messages are delivered to the carrier over a network connection.

To deliver SMS messages, adTempus must connect to an SMS server. Two kinds of servers are available:

- Your company may operate or subscribe to an SMS gateway service. The gateway accepts SMS messages and automatically routes them to the proper carrier for delivery to the device. That is, messages for both your SkyTel pager and you Cingular cell phone can go to the same gateway.
- The carrier may provide public access to its SMS server, allowing adTempus to connect directly to it. When you use this option, messages for each device must be sent by adTempus directly to the SMS server for the company that provides service for the device. That is, messages intended for your SkyTel pager must be delivered to the SkyTel SMS server; messages intended for your Cingular cell phone must be delivered to the Cingular SMS server.

Finding the Proper Settings

To configure the service provider, you will need to know the network address that adTempus should connect to, and any user information needed to connect to the server.

If you are using a gateway service, check with that service for more information.

If you want to connect directly to the carrier, you will need to check with the carrier for the connection information. adTempus has a list of carriers whose public servers we know about; you can check to see if your carrier is on that list by using the Import Settings... button.

Related Topics

Messaging Provider Import 425

Reference

SMS Service Provider Properties 422

SMS Service Provider Properties

Location: Available from the Messaging Setup Window

The **SMS Service Provider Properties** window contains the settings for an SMS Service Provider.

Properties

Provider Page

This page defines general options for connecting to your SMS gateway.

adTempus has a list of settings for some carriers and gateways. Try the **Import Settings** button and see if you carrier is listed. If not, see the Overview topic for information on how to find the settings you need.

Provider Name

Provide a descriptive name for the carrier or gateway service.

Enabled

Check this box to enable the provider. If the box is unchecked the provider is disabled and messages will not be sent using it.

Server

Specify the name or address and port number for the SMS server. Enter the System Type if specified by the provider.

User ID

Specify the user ID/caller ID used when connecting to the SMS server.

Password

Specify the password used when connecting to the SMS server.

Maximum Message Length

Most SMS providers specify a limit for the length of messages. Messages that exceed the limit may be rejected. Specify the maximum here, and adTempus will truncate messages that exceed the length.

Automatically split messages that exceed the maximum

When this option is checked, adTempus will split notification messages that exceed the maximum length into separate messages, each shorter than the limit. This allows adTempus to deliver the complete notification message to your SMS device, as a series of messages.

You can set an absolute maximum length for messages by using the limit on the Notification Address.

Import Settings

Click this button to import provider settings from a list of known providers. See the Import topic for more information.

Test

Click the **Test** button to send a test message using this provider.

Security Page

The Security page contains a standard Security Editor that allows authorized users to manage the security for this Provider. Permission to create new providers is controlled through the Notification Recipient Security options.

Related Concepts

SMS Service Provider 421

Related Topics

Messaging Setup 417
Messaging Provider Import 425

Jabber Service Provider Properties

Location: Available from the Messaging Setup Window

The Jabber Service Provider allows you to send notification messages using the Jabber instant messaging protocol. This is the protocol used by Google Talk.

For more information on Jabber, Jabber clients, and Jabber servers, see www.jabber.org.

Once you have configured a provider, you can create Notification Addresses that use it.

Property Pages

Provider

Provider Name

Provide a descriptive name for this provider.

XMPP Server

Enter the DNS name or IP address of the server. To use the Google Talk server, enter `gmail.com` as the server name.

Port

Enter the port number to use, or leave blank to use the default port.

User ID and Password

Specify the user ID and password used to log in to the Jabber server. For example if you are using Google Talk and your Gmail address is `example@gmail.com`, the User ID is `example@gmail.com`.

Test

Click the Test button to send a test message using this provider.

Security

The Security page contains a standard Security Editor that allows authorized users to manage the security for this Provider. Permission to create new providers is controlled through the Notification Recipient Security options.

Related Topics

Messaging Setup 417

Messaging Provider Import

Arcana Development maintains a list of SMS gateway providers that we know about. If yours is on the list, you can import the settings when setting up the provider in adTempus.

To check the list, click the **Import Settings...** button in the SMS Service Provider Properties window.

In the **Messaging Provider Import** window, select the **Download the latest list** option and click **Load**. (If your computer is not connected to the Internet, you can use the **Load from a file** option to load providers from a file installed with adTempus.)

Then look for your provider in the **Provider to Import** list and click **OK** once you find it.

If your provider is not on the list, you will need to configure the provider yourself.

Related Concepts

SMS Service Provider 421

Reference

SMS Service Provider Properties 422

Notification Recipient Security

The **Notification Recipient Security** page is used to manage default security settings for Notification Recipients and Notification Groups, and control the ability to create new objects of these types. Permissions set here are inherited by all of those objects

See the Security Editor topic for more information on editing security settings.

The following permissions apply to notification recipients:

Permission	Description
Full Control	Permission to perform all actions on the recipient.
List/Use	Permission to use the recipient.
View	Permission to view the properties of the recipient.
Create	Permission to create new objects.
Modify	Permission to modify the properties of the recipient.
Delete	Permission to delete the recipient.
Administer security	Permission to change the security settings for the recipient.
Change owner	Permission to take ownership of the recipient.

Object Comparison Report

Location: Available from the Object Change Log

The **Object Comparison Report** displays a comparison of the settings between two versions of an object, based on snapshots of the object.

For each setting in the object, the report lists the value from each of the two versions.

Related Concepts

Auditing and Snapshots 364
Snapshot Overview 364

Object Configuration Report

The **Object Configuration Report** provides a quick listing of all of the settings of an object, which can be used to document the state of the object or quickly review settings.

The Object Configuration Report is not located in the Reports view with other reports. Instead, you can display the configuration report for an object by right-clicking it and choosing the **Configuration Report** command from the context menu.

From the Configuration Report window you can save or print the report.

Queue Security

The Queue Security window is used to manage the default security settings for Job Queues, and controls permission to create new Queues. Permissions set here are inherited by all Job Queues. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Job Queues:

Permission	Description
Full Control	Permission to perform all actions on the Queue.
List/Reference	Permission to list this Queue.
View	Permission to view the properties of the Queue.
Create new Queues	Permission to create new Queues.
Associate (assign jobs to this queue)	Permission to assign jobs to a Queue.
Modify	Permission to modify the properties of the Queue.
Delete	Permission to delete the Queue.

Permission	Description
Administer security	Permission to change the security settings for the Queue.
Change owner	Permission to take ownership of the Queue.

Remote Agent Security

This window controls the default security for Remote Agents and controls the ability to create new Remote Agents. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Remote Agents:

Permission	Description
Full Control	Permission to perform all actions on the agent.
List/Use	Permission to schedule jobs to run on this agent.
View	Permission to view the properties of the agent.
Create	Permission to create new agents.
Modify	Permission to modify the properties of the agent.
Delete	Permission to delete the agent.
Administer security	Permission to change the security settings for the agent.
Change owner	Permission to take ownership of the agent.

Resolve Object References Window

The **Resolve Object References** window is displayed when you try to delete an object that is being used elsewhere in adTempus. For example, if you try to delete a Notification Recipient that is being used in a job, adTempus will display the Resolve Object References window.

Before the object can be deleted, the references must be replaced or removed. For example, if the Notification Recipient is used in a Notification Action in the job, the Recipient must be removed from the list of recipients on that action, or replaced with a different recipient.

You may edit references individually to replace or remove the references, or select the appropriate action and click **Replace Selected References** to apply modifications in bulk.

When you click **OK**, adTempus will check again to see if any references are found. If so, this window will be displayed again. Otherwise the deletion operation will continue.

Clicking **Cancel** aborts the deletion operation but does not undo any changes that have already been made.

Related Topics

Object References Window 443

Script Security

This window controls the default security for Shared Scripts and Script Libraries, and controls the ability to create objects of these types. Permissions set here are inherited by all Shared Scripts and Script Libraries.

See the Security Editor topic for more information on editing security settings.

The following permissions apply to scripts and script libraries:

Permission	Description
Full Control	Permission to perform all actions on the script or library.
List/Use	Permission to use the script or library.
View	Permission to view the properties of the script or library.
Create	Permission to create new scripts and script libraries.
Modify	Permission to modify the properties of the script or library.
Delete	Permission to delete the script or library.
Administer security	Permission to change the security settings for the script or library.
Change owner	Permission to take ownership of the script or library.

Send Error Logs Window

Location: **Help > Send error logs...**

This window allows you to send diagnostic log files from the adTempus server to Arcana Development for troubleshooting.

You can configure adTempus to send these logs automatically using the settings on the Data Collection Page of the General Server Options window.

While the diagnostic logs will never contain passwords stored in Credential Profiles in adTempus, they may contain other information that is not anonymous, such as command-line parameters, computer names, IP addresses, e-mail addresses, program names, etc.

The log files are compressed and encrypted before being transmitted to Arcana Development, and are treated as confidential information once they are received.

Sequence Jobs Tool

Location: From the main Console Tree, right-click a Group and select **Sequence jobs in group**

Jobs are linked together in adTempus using Job Control Actions in Responses. To create a sequence of jobs, each job has a Response that runs the next job in the sequence.

The **Sequence Jobs Tool** provides a shortcut for creating and reordering such links.

When you open the tool for a Job Group, all jobs for that group and, optionally, its sub-groups, are listed. Check the boxes next to the jobs that you want to have participate in the sequence, and use the up and down arrows to change the order in which the jobs will run.

After setting options, click **OK** to apply the changes.

This tool does not affect any Job Control Actions that target jobs that have not been selected for the sequence.

Add sequence number to job names

When you check this option, adTempus will add a sequence number prefix to the name of each selected job. For example, if you have selected three jobs for your sequence, named "Job A," "Job B," and "Job C," adTempus will rename them to "1. Job A," "2. Job B," and "3. Job C."

You can change the starting number for the first job in the sequence, and the amount by which the number is incremented for each job.

Link type

Select the kind of rule that will be created for the links that join the jobs:

- **On Success.** The next job is run only if the current job succeeded.
- **On Completion.** The next job is always run when the current job finishes, regardless of whether it succeeds or fails.

If you check the **Preserve existing link type** option, adTempus will keep the current rule, if any, defined for each job.

Server Connection

The **Server Connection** window specifies properties for a new or existing adTempus server connection.

These options only control how the adTempus Console connects to the adTempus Server. To set options for the server, see the Server Options topic. Additional user-specific settings that apply to all server connections can be found in the Console Options window.

To connect to a new server, select the **Connect Server...**command from the **File** menu in the adTempus Console. To view or modify settings for an existing connection, right-click the server's name in the Console Tree and select **Connection Properties**. Use the other commands on this menu to disconnect a server or to remove it from your connection list.

Target

If you are establishing a new connection, specify the server you want to connect to. Select the local computer or enter the name or address of a remote computer.

Instance

If you are connecting to an instance of adTempus other than the default, enter the instance name.

Port

If the adTempus server is configured to use a communications port other than the default, enter the appropriate port value.

Connection Name

Enter a name for this connection. This is the name that will be used to identify the server in the Console Tree. The name can be the same as the server name, or any other name you wish.

Authentication

The Authentication option determines how the adTempus service will identify you.

Windows Authentication

When you use Windows Authentication, you do not need to specify a user ID and password. adTempus identifies you based on your Windows login.

adTempus Authentication

In situations where Windows authentication cannot be used, your adTempus login may have been configured using adTempus authentication, which requires a user ID and password.

Check the **Save password** option to have your password saved in the connection settings. If you do not check this option, you will be prompted for the password each time you connect to this server.

Prompt when connecting

Choose this option to have adTempus prompt you for login information each time you connect to this server.

Automatically reconnect whenever the Console is opened

If this option is checked the adTempus Console will try to reconnect to this server each time you start the Console. If this option is not checked, the server will still be listed in the Console, but adTempus will not try to connect to the server until you expand its node.

Automatically expand the server's node in the Console Tree

Check this option to have the Console expand this server's node after automatically connecting it.

Make this my default server

You may select one server connection to be your default server. You can use the **Startup View** setting in the Console Options Window to have the Console go directly to a view within your default server each time you start the Console.

Server Options

Server Options Window

Location: **Configuration > Server Options > General Options**

The Server Options window contains server-level settings for adTempus.

The security settings found in this window in earlier version of adTempus are now found in the Server Security Settings window.

Settings Pages

The settings are divided among the following pages:

General

Data Backup

Auditing and Snapshots

Variables

Data Collection and Error Reporting

Additional Options

View server settings change log

This button opens the Change Log window for the server settings. The auditing and snapshot settings for the server determine whether audit records and snapshots are available for the server settings.

Advanced Options

This button opens the Advanced Server options window.

Related Topics

Messaging Setup ... 417
File Servers Window ... 397
Server Security Settings ... 437
Linked Servers Window ... 414
Console Options Window ... 386

General Page

Location: Server Options Window

Data Retention

The **Data Retention** section contains settings that determine how long some data is kept by adTempus.

Default History Retention

The Default History Retention option determines how long adTempus retains the history for jobs when the job's history retention option is set to "Use global default setting." History information is removed from adTempus once it is older than the retention period.

By default adTempus is configured to keep information on the most recent 30 executions of each job. If you change this option to retain history for a certain number of days, be sure to consider the effect on jobs that run frequently, and adjust the settings for those jobs to use a shorter retention period as appropriate.

For example, suppose you change the default retention to 60 days, and you have some jobs that run every 5 minutes. These jobs will generate 17,280 instances in a 60-day period. Having a such a large amount of history for many jobs can create performance and database size issues for adTempus depending on your database configuration.

Retain server messages (such as unacknowledged alerts) for __ days

Most messages in the adTempus message log are related to a particular job; these messages are purged from the database according to the history retention settings for the job.

Some messages are not related to any job; these are purged after the number of days you specify here.

Alerts that have not been acknowledged are not deleted even if they are older than the age limit.

Do not delete failed instances that have not been acknowledged

When this option is checked, adTempus will never remove failed job instances that have not been acknowledged, even if the retention period has expired for them. If it is not checked, such instances will be removed according to the retention rules.

Database Operation Task

Timeout for Database Operation Task Commands

This value determines how long database operations executed by the Database Operation Task are allowed to execute. The default value is 20 minutes; increase this value for operations that require more time. The timeout applies only to select and update operations; database job execution is not subject to a time limit.

Data Backup Page

Location: Server Options Window

The Data Backup Page contains settings that determine how adTempus data is backed up. Two forms of backup are available, and each serves a slightly different purpose. We recommend that you create both a snapshot and a database backup each day.

Create a daily snapshot of all adTempus data

This option creates a daily snapshot of the adTempus data. Snapshots allows a granular restoration of individual objects (for example, you can undo the changes for a single job, or recover a deleted script). They also allow you to view and compare changes between versions of objects. However, they do not include job history data, and are less useful than database backups in the event of a catastrophic failure that results in the loss of your adTempus database.

System snapshots appear in the Change Log for each object contained in the snapshot.

Members of the adTempus Administrators group can create an on-demand system snapshot using the Create Snapshot command.

Create a daily backup of the adTempus database

This option creates a daily backup of the adTempus database. Database backups allow you to recover quickly in the event of a database or system failure. They include job configuration and job history.

If the database server that adTempus is using is located on the same computer as adTempus (as is the case if you are using the default SQL Server Express database), the database backup will be located on that computer as well. You should copy the database backup file to another computer each day to protect it in the event of a system failure.

See the Backing Up and Restoring adTempus Data topic for more information on database backups.

Back up and snapshot each day at

Set the time at which the backup and snapshot should run.

Auditing and Snapshot Settings

Location: Server Options Window

The settings on this page allow you to configure Auditing and Snapshots for adTempus.

If you want to be able to compare versions of an object, or restore a modified or deleted object, be sure to turn on snapshots for the relevant object types (see below).

Configuration

For each type of object in adTempus you specify which change tracking is applied to each event.

Auditing

The following audit events can be configured for each kind of object:

- **Create**
- **Modify**
- **Delete**
- **Enable/Disable** (for Jobs, Queues, and Groups, this applies to Release and Hold operations)

The following settings are available for audit events:

- **None.** No audit entry is recorded
- **Automatic.** An audit entry is always recorded, with no input from the user.
- **Comments Optional.** An audit entry is always recorded. The user is prompted to optionally enter comments about the change. The user does not have to enter comments, but cannot suppress the audit entry.
- **Comments Required.** An audit entry is always recorded. The user is prompted to enter comments about the change. The user is required to enter comments, and cannot suppress the audit entry.

Snapshots

The following snapshot events can be configured for each kind of object:

- **Modify.** The snapshot is taken before the change is applied. The snapshot will appear in the Change Log for the object.
- **Delete.** The snapshot is taken before the object is deleted. The object can be restored from the snapshot using the Deleted Objects command.

You must have deletion snapshots turned on (set to **Automatic**, **Comments Optional**, or **Comments Required**) to be able to restore deleted objects.

The following settings are available for snapshot events:

- **None.** No snapshots are created. The user is not permitted to create an on-demand snapshot.
- **On-Demand.** No automatic snapshots are created, but the user may create an on-demand snapshot.
- **Automatic.** A snapshot is always recorded, with no input from the user.
- **Comments Optional.** A snapshot is always created. The user is prompted to optionally enter a label and comments for the snapshot.
- **Comments Required.** A snapshot is always created. The user is prompted to enter comments and an optional label for the snapshot.

Retention Settings

The retention settings determine how long audit records and snapshots are kept.

- **Retain audit records.** Determines how long the audit record is kept. The audit record notes who made the change, but does not contain information about what specific settings were changed.
- **Retain snapshots.** Determines how long snapshots are kept, for actions other than Delete.
- **Retain snapshots of deleted objects.** Determines how long snapshots of deleted objects are kept. The snapshot can be used to recover a deleted object. Once the snapshot has been purged, the object cannot be recovered.

These settings do not apply to automatic full-system snapshots created as part of the daily backup process, which have their own retention setting on the Data Backup page.

Related Concepts

Snapshot Overview....364
Auditing and Snapshots....364

Related Topics

Window Reference....384
Object Comparison Report....426

Variables Page

Location: Server Options Window

The **Variables** page allows you to define Job Variables for this server. Any Variables you define here are inherited by all Job Groups and Job Queues (and through them, Jobs) on the server.

Data Collection and Error Reporting Page

Location: Server Options Window

The **Data Collection** page contains options for reporting adTempus usage information to Arcana Development.

Anonymous usage information

Check this option to allow adTempus to report information about what features are being used in adTempus. This information helps us understand how customers are using the software. The information is collected and reported anonymously. It does not include any information from your adTempus data, just counts of the number of times you use particular features.

Note: This option controls feature usage reporting on the adTempus server. Feature usage reporting for the adTempus Console is configured separately for each user, through the Console Options window.

Diagnostic logs

Check this option to allow adTempus to automatically send log files from the adTempus server to Arcana Development when errors are detected.

adTempus writes to the diagnostic logs (found in the "logs" subdirectory under the adTempus program directory) when unexpected problems occur within adTempus. These problems may be due to programming errors or to unexpected scenarios or environmental conditions. In some cases these problems may result in an error message being reported to you in the Alerts view or the Job Log. Sending these logs to Arcana Development helps us to proactively identify problems in adTempus.

While the diagnostic logs will never contain passwords stored in Credential Profiles in adTempus, they may contain other information that is not anonymous, such as command-line parameters, computer names, IP addresses, e-mail addresses, program names, etc.

The log files are compressed and encrypted before being transmitted to Arcana Development, and are treated as confidential information once they are received.

If you prefer not to use automatic submission, you can submit diagnostic logs manually using the Send error logs command on the Help menu, after reviewing the contents of the log files for sensitive information. adTempus will log an Alert when it detects that errors have been recorded, prompting you to send submit the logs.

Error Notifications

The **Configure alert notifications** button opens the Alert Notification Configuration Window, where you can configure adTempus to send notification problems when adTempus encounters errors or operational problems.

Advanced Server Options

Location: In the Server Options Window, click the **Advanced Options** link

The Advanced Server Options window provides access to advanced configuration options for adTempus that generally do not need to be modified. You should make changes to settings in this window only with guidance from support staff.

Server Security Settings

Location: **Configuration > Security Settings**

The **Server Security Settings** window allows adTempus Administrators to manage adTempus logins and security groups and to configure top-level security settings for adTempus.

Security Settings

The **Security Settings** page defines permissions that are inherited by all objects in adTempus. Any permissions you assign here will apply to all jobs, groups, queues, and other objects throughout adTempus. For example, by default the "All Users" group is granted "View" permission here, which allows all adTempus users to view all objects in adTempus. See the Security Inheritance topic for information on how to assign permissions at a more granular level (for example, to grant a group of users permission for a specific set of jobs).

Apply Permissions To

This option determines which objects the selected security entry applies to:

- **Server settings, all objects.** The permissions apply to the server settings themselves (e.g., permission to modify server settings) and to all other objects in adTempus.
- **All objects below this level.** The permissions apply to all objects in adTempus except the server settings.
- **Server settings only.** The permissions apply only to the server settings and are not inherited by other objects.

Logins

The **Logins** page lists all logins (users) defined for adTempus. Users with the required level of access can add, modify, or delete logins.

Include dynamic logins

Check this option to include dynamic logins (logins created automatically for users based on their membership in Active Directory security groups).

Groups

The **Groups** page lists all security groups (roles) defined for adTempus. Users with the required level of access can add, modify, or delete groups.

Related Topics

Server Options 431
Messaging Setup 417
File Servers Window 397
Linked Servers Window 414
Console Options Window 386

Shared Schedule

A shared schedule is a schedule that can be used by more than one job. For example, you might have a predefined "weekdays" schedule for jobs that executed Monday through Friday. When you create a new job that needs to run on weekdays, you can link it to this shared schedule, rather than having to define a new schedule for the job.

Note that the shared schedule only defines the days on which jobs should run; each job has its own settings to determine the time(s) at which it should run.

Shared schedules are used on the Date Selection page of a Schedule.

To view or manage shared schedules, select the Shared Schedules folder in the adTempus Console.

Use the settings on the **Security** page to specify which users are allowed to use this schedule.

Related Topics

Shared Schedules 381

Reference

Shared Schedule Properties 439

Shared Schedule

A shared schedule is a schedule that can be used by more than one job. For example, you might have a predefined "weekdays" schedule for jobs that executed Monday through Friday.

When you create a new job that needs to run on weekdays, you can link it to this shared schedule, rather than having to define a new schedule for the job.

Note that the shared schedule only defines the days on which jobs should run; each job has its own settings to determine the time(s) at which it should run.

Shared schedules are used on the Date Selection page of a Schedule.

To view or manage shared schedules, select the Shared Schedules folder in the adTempus Console.

Use the settings on the **Security** page to specify which users are allowed to use this schedule.

Related Topics

Shared Schedules 381

Reference

Shared Schedule Properties 439

Shared Schedule Properties

The **Shared Schedule Properties** window contains the settings for a Shared Schedule.

Property Pages

Schedule

The **Schedule** page defines the days on which jobs that use this shared schedule should execute.

Name

Provide a descriptive name for this schedule. The name must be unique across all shared schedules and holiday sets.

Enable this schedule

Check the box to enable the schedule or clear the box to disable it. If a shared schedule is disabled, it will not appear in the list of available shared schedules when you are creating a job.

⚠ If the schedule is disabled, jobs that use it will not be triggered by the schedule.

Rules for all years

The rules in this section will apply to all years.

Date rules can be specified using either a simple interval or more complex rules.

★Trigger every ___ days

Specify the interval (in days) at which the item should be triggered. The interval is calculated from the specified starting date.

Trigger on specific days

Specify any number of specific dates or date rules to define the days on which the item will be triggered. See the Select Days topic for information on specifying days.

Rules for specific years

The rules in this section allow you to create Day Selection rules that apply only to specific years or date ranges.

Notes

Optionally enter an extended description or notes for this schedule.

Security

The **Security** page is used to view or modify the security settings for this object. See the Security Editor topic for more information on editing security settings.

The following permissions apply to Shared Schedules:

Permission	Description
Full Control	Permission to perform all actions on the schedule.
List/User	Permission to use the schedule.
View	Permission to view the properties of the schedule.
Modify	Permission to modify the properties of the schedule.
Delete	Permission to delete the schedule.
Administer security	Permission to change the security settings for the schedule.
Change owner	Permission to take ownership of the schedule.

Permission to create new Shared Schedules is controlled through the Shared Schedule and Holiday Security window.

Related Concepts

Shared Schedule 438

Related Topics

Shared Schedules 381

⭐Shared Schedule and Holiday Security

This page sets the default security options for Shared Schedules and Holiday Sets, and controls the ability to create objects of these types. Permissions assigned here are inherited by all Shared Schedules and Holiday Sets.

See the Security Editor topic for more information on editing security settings.

The following permissions apply to Shared Schedules and Holiday Sets:

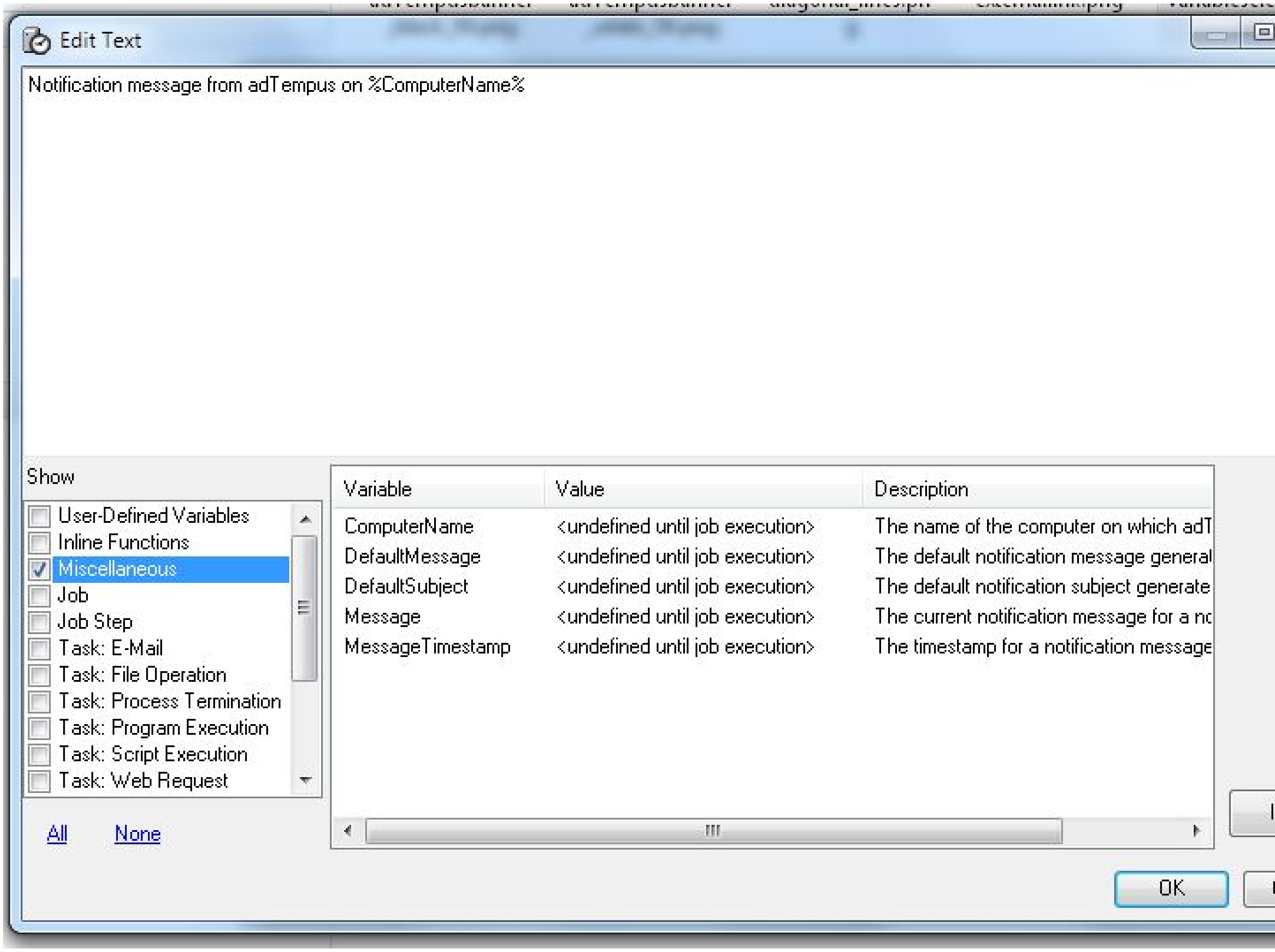
Permission	Description
Full Control	Permission to perform all actions on the schedule or holiday set.
List/User	Permission to use the schedule or holiday set.
View	Permission to view the properties of the schedule or holiday set.
Create	Permission to create new schedules and holiday sets.
Modify	Permission to modify the properties of the schedule or holiday set.
Delete	Permission to delete the schedule or holiday set.
Administer security	Permission to change the security settings for the schedule or holiday set.
Change owner	Permission to take ownership of the schedule or holiday set.

Text Edit Window

This feature is new for adTempus version 4.

The **Text Edit Window** provides a pop-up editing window for many text fields throughout the adTempus Console. It serves the following purposes:

- Provides more space for editing text.
- Allows you to select Job Variables and Inline Functions from a list and insert references to them in your text.



Notification message from adTempus on %ComputerName%

Show

- ☐ User-Defined Variables
- ☐ Inline Functions
- ☑ Miscellaneous
- ☐ Job
- ☐ Job Step
- ☐ Task: E-Mail
- ☐ Task: File Operation
- ☐ Task: Process Termination
- ☐ Task: Program Execution
- ☐ Task: Script Execution
- ☐ Task: Web Request

Variable	Value	Description
ComputerName	<undefined until job execution>	The name of the computer on which adT
DefaultMessage	<undefined until job execution>	The default notification message general
DefaultSubject	<undefined until job execution>	The default notification subject generate
Message	<undefined until job execution>	The current notification message for a nc
MessageTimestamp	<undefined until job execution>	The timestamp for a notification message

To open the Text Edit window, click the Insert Variables (▤) button, which is found next to any text entry box that supports Job Variables.

At the bottom of the editor window you find a list of all defined Job Variables and Inline Functions, grouped by category. To insert a variable into your text, position the cursor at the desired location, highlight the variable name, and click **Insert**.

adTempus has many system-defined variables, which get set during job execution by various adTempus components. These are categorized in the list based on the component that sets them

The "User-Defined Variables" category includes all variables that have been defined by users, at the Server, Group, Queue, Job, Step, etc., level.

If your job creates new variables at runtime through scripts or through the Variable Update Action or Variable Update Task, they will not appear here, since the Editor does not have any way of knowing about them.

Related Concepts

Job Variables 88

Reference

Job Variable Properties 412

Object References Window

The **Object References** window shows you all the references to an object within adTempus. For example, you can use this tool to find all the places where a particular Notification Recipient is used.

This tool is available using the **Find/Replace References** menu command in the following places:

- Notification Recipients Folder
- Shared Schedules Folder
- Holiday Definitions Folder
- Shared Scripts Folder
- Script Libraries Folder
- Remote Agents Folder
- Credential Profiles Window
- Messaging Setup Window
- File Servers Window

To find links between jobs, use the Job Diagram tool instead. To find text within fields such as job names, notes, command lines, etc., use the Find and Replace tool.

The window will list all the places where the object is being used. Click the Location link in the list to view or edit the item that references the object.

You can also remove or replace references (for example, modify jobs that use the Credential Profile for account "Bob" to use the Credential Profile for account "Claire" instead.

Related Topics

Find and Replace 460
Resolve Object References Window 427

Tools Reference

Data Import and Export

The adTempus Import/Export facility allows you to copy data between adTempus servers. The following kinds of objects can be transferred:

- Jobs
- Shared Schedules
- Holiday Sets
- Messaging Service Providers
- Notification Recipients
- File Service Providers
- Shared Scripts
- Script Libraries
- Server Settings

When you export an object, any objects on which it depends are automatically exported. For example, when you export a job, any scripts, notification recipients, shared schedules, etc., that are used in the job are exported with it.

When you import data from a data file, adTempus checks each object to see if it already exists. If it does, the object already in adTempus is replaced if the object in the file has a later timestamp.

adTempus 4.0 and later use a different export format than previous versions of adTempus, but is able to read and write the format used by version 3.

adTempus 3.0 and later use a different export format than previous versions of adTempus. adTempus 3.0 and later cannot import data exported from a version older than 3.0, and vice-versa.

Data Format

adTempus exports data in XML format. However, this file is not designed to be used to import data from other scheduling systems or to be created from scratch to load data into adTempus. If you need an automated method to create and configure jobs, you should use the adTempus API instead.

Related Topics

Data Export 445
Data Import 446

Data Export

Location: **Tools > Import/Export > Export**

The Data Export tool allows you to export data from adTempus. You can export data to a file to be saved for backup or to be transferred to another adTempus server. You may also export the data directly to any other server to which the Console is currently connected.

You can only export data for which you have Modify permission.

If you are trying to recreate your adTempus instance in a new database, it is more thorough and more efficient to do this by copying the entire database rather than exporting and importing all of the data. For information on how to copy the entire database, refer to this Knowledge Base article.

Object Selection

In the Data Export window, select an object type to view a list of objects of that type, then select the objects you wish to export. You can export objects of multiple types at the same time (e.g., you can export jobs and shared schedules).

When you export an object, any objects on which it depends are automatically exported. For example, when you export a job, any scripts, notification recipients, shared schedules, etc., that are used in the job are exported with it. Thus if you want to export a single job, that is the only object you need to select.

adTempus 4.0 and later use a different export format than previous versions of adTempus, but is able to read and write the format used by version 3.

adTempus 3.0 and later use a different export format than previous versions of adTempus. adTempus 3.0 and later cannot import data exported from a version older than 3.0, and vice-versa.

Options

Include history when exporting jobs

When this box is checked, adTempus will include the history (instance details and log messages) for all exported jobs. Otherwise, history will not be included.

You must check the corresponding box when you import the data, or the history data in the file will be ignored by the import process.

Optionally you may specify a date/time range for the history to be included.

Include passwords for credential profiles

When this box is checked, adTempus will include the passwords for any Credential Profiles associated with jobs being exported. The passwords will be stored in an encrypted format.

If this box is not checked, passwords are not included, and you will be prompted for the password for each Credential Profile when the data is imported.

Specify a password to protect sensitive data

Sensitive data (like passwords) is encrypted when it is written to the export file. By default adTempus uses a standard encryption key that allows the file to be read by any other adTempus installation. This means that although the data is protected from people who casually view the export file, it is possible for anyone with a copy of adTempus to read and view the data.

For maximum protection you can provide your own password to use when encrypting data. The same password will be required when the data is imported.

Comments

You may optionally enter comments that will be stored in the export file. The comments are displayed when the file is selected in the Import Wizard, and can also be seen by viewing the file with a text or XML viewer.

Export Target

Export to File

Select this option to save the exported objects to a file. See the import/export overview topic for more information on the file format.

File Format

Select the format for the export file. If you select the "Version 3 compatible" option, the file can be used to transfer data to an adTempus version 3 installation, but adTempus will not be able to export objects or settings that are not supported by version 3.

Export to Server

Select this option to transfer the objects directly to another server. Only servers to which the Console is currently connected will be listed.

Related Concepts

Tools Reference 444

Related Topics

Data Import 446

Data Import

Location: **Tools > Import/Export > Import**

The Data Import wizard allows you to import data that has been exported from this or another adTempus server using the Export tool.

adTempus 4.0 and later use a different export format than previous versions of adTempus, but is able to read and write the format used by version 3.

adTempus 3.0 and later use a different export format than previous versions of adTempus. adTempus 3.0 and later cannot import data exported from a version older than 3.0, and vice-versa.

The Import Wizard comprises the following steps:

1. Select Source
2. Select Objects
3. Import Options
4. Import Results

Related Concepts

Tools Reference 444

Related Topics

Data Export 445

Source Page

The **Source** page of the Import Wizard allows you to select the source for the imported data, and specify import options.

Import data from file

Specify the file to import from. This file must have been created using the adTempus export tool. See the import/export overview topic for more information on the file format.

adTempus 3.0 and later use a different export format than previous versions of adTempus. The import tool can only read files created by adTempus version 3.0 and later.

It is possible to transfer data directly between adTempus servers, without exporting to a file and importing. To do this, use the **Export to Server** option in the Export tool.

When you select a file, any comments saved in the file during export will be displayed below the file name.

Next Step

When you click **Next**, adTempus will read the file you have specified and make sure it is valid. If it is not, you will not be able to continue.

If the file is valid, the Wizard will continue to the Select Objects step.

Related Concepts

Tools Reference 444
Data Import 446

Select Objects Page

The **Select Objects** page of the Import Wizard lists all objects found in the source file you have selected. For each object, the State is one of the following values:

- **New.** This is a new object.
- **Existing.** This object already exists in adTempus. The data from the file will replace the current data in adTempus.
- **Duplicate.** Another object with the same name already exists in adTempus, but it has a different identity (see More About Duplicate Objects, below. The data from the file will replace the current data in adTempus.

Select the objects that you want to import by checking the box next to them.

Note: Some objects may be imported even if you don't select them in the list, if they are referenced by objects that you have selected to import. For example, suppose you have selected to import Job "Job A," and Job A uses Credential Profile "freduser." Because this Credential Profile is needed by Job A, it will be imported even if you don't select it in the list.

Don't import referenced objects that already exist on this server

When this option is checked, adTempus only shows objects that were specifically selected when the export was performed. Other objects included in the file are not shown, and are not imported if they already exist on the server. If they do not already exist on the server, they will still be imported.

When you export a job from adTempus, you select just the job in the export window. However, adTempus also exports all other objects that the job links to or depends on. This includes

- The Job Group that the job is in (and all parent groups of that group)
- The Job Queue
- The Credential Profile
- Any Notification Recipients, Shared Scripts, or other shared objects the job uses
- Any jobs the job is linked to through Responses, Conditions, or Triggers

All of this data is exported so that the job (and its chain) can be fully recreated on the target server. If these other objects already exist on the target server, the Select Objects page indicates this (see above) and lets you decide which objects should be imported. When you check **Don't import referenced objects that already exist on this server**, these other objects are not shown in the list, and are not imported unless they don't already exist. This makes it easier to focus on only the data that was selected for export.

For example, suppose you are exporting a job from your Test environment and importing it into your Production environment. You only want to import new settings for the job, not for its Job Group, Queue, etc. When **Don't import referenced objects that already exist on this server** is checked, these objects don't appear in the object selection list, so you don't have to worry about importing them accidentally and overwriting the production settings for these objects.

This option is only available if the export was created by adTempus 4.3 or later.

When you export data, adTempus marks in the export file which object(s) were selected in the Export window, and this information is used to filter out objects that were not explicitly selected.

Special Considerations for Server Settings and Security Containers

If you chose to export server settings, the object list may contain the following special items:

- **Server Settings:** Select this item if you want to import general server options, server-level Job Variables, or audit and snapshot settings. You will be able to choose which of these to import on the Import Options page.
- **Security Container:** If you exported default security settings for object types, they will be listed here as, for example "Security Container: Default Job Group Security." If you import these items, they will **replace** the security settings for the corresponding security containers. See the Security Inheritance topic for information about what each of these containers affects.

If you choose any Security Containers, you must also check the **Import security settings** option on the Import Options page.

Use care when importing Server Settings and Security Containers, as these settings can have far-reaching consequences. You should create a database backup or system snapshot before importing these items.

Next Step

When you have finished making your selections, click **Next** to continue to the Import Options step.

More About Duplicate Objects

Each object in adTempus has an internal identifier (or "key") that you do not see. This identifier never changes, even if the object's name or other properties change. This allows adTempus to always uniquely identify each object.

If you export an object and then re-import it into the same adTempus server, the identifier in the import file will match the object's identifier in the database, so adTempus will know that the object is "Existing." This is true even if you have changed the name of the object. For example, suppose that you export job "Job A" to a file, then rename it to "Job B." When you import that file, adTempus will know that "Job A" in the file is the same object as "Job B" in adTempus. If you choose to import the job, the values from the file will replace the values currently in adTempus.

However, support you independently created jobs named "Fred's Job" on Server A and Server B. Since these jobs were created separately, they will have different internal identifiers even though they have the same name. If you now export "Fred's Job" from Server A and import it to Server B, adTempus will detect that there is already a job named "Fred's Job," but the existing job has a different internal identifier. In this case adTempus will list the state of "Fred's Job" as "Duplicate," indicating to you that there is already another job with that name.

When a duplicate object is found, there are two possibilities:

- You want the object in the file to replace the object already in adTempus. In this case, simply check the box to import the object.
- These are really two separate objects, and you don't want the existing object to be replaced. For example, suppose "Fred's Job" from Server A does something completely different from the "Fred's Job" on Server B, and you want to keep both of them. In this case you need to cancel the import and either
  - Rename the object on the original server and re-export it, or
  - Modify the import file to rename the object there. See the import/export overview topic for more information on the file format.

Related Concepts

Tools Reference 444
Data Import 446

Import Options Page

The **Import Options** page of the Import Wizard allows you to set options for the import.

Basic Options

The following options

Import job history if present in file

When this option is checked, adTempus will import any history records found in the file for imported jobs. If this option is not checked, history will be ignored. Note that history data will not be available unless the option to export history data was checked when the data was exported.

When you select this option, the **Save all imported objects if no problems are found** option will automatically be checked and you will not be able to review imported data before saving it. If any errors occur during the import and you are prompted to review objects and

⭐save them manually (for example, if you are prompted to supply the password for a Credential Profile), the job history will not be saved. In this case, rerun the import (selecting only the jobs you need want to import history for). Since all problems were resolved on the first try, the history import should succeed this time.

Import security settings for objects

When this option is checked, adTempus will import the security settings for each object. If the option is not checked, the security information in the export file will be ignored, and imported objects will receive the same security settings as if they had been newly created through the Console.

Save all imported objects if no problems are found

When this option is checked, adTempus will attempt to save all imported objects if no problems are found.

If this option is not checked, or if problems are found with the imported data, the objects will not be saved automatically, and you can review them from the Import Results page before saving them.

Generate comparison report for existing objects

When this option is checked, adTempus will create a comparison report for each object that is in the import source and that already exists in the adTempus database. The reports will be available on the Import Results page, allowing you to compare the two versions of each object.

Create snapshots of existing objects before updating them

When this option is checked, adTempus will create a Snapshot of each object that is updated by the import. This will allow you to revert to the previous version of the object (undo the effects of the import) if necessary.

Show advanced options

Check this option to display the Advanced Options (see below) when you click **Next**.

Server Settings Import

The Server Settings Import options are available only if you have chosen to import Server Settings (on the Select Objects page). These settings determine which portions of the server settings are imported:

- **General server settings:** Includes server configuration settings not covered by other categories, such as backup and alerting settings, history retention settings, etc.
- **Server-level job variables:** Includes server-level Job Variables defined in the Server Options window.
- **Audit and snapshot options:** Includes the settings from the Audit and Snapshots page of the Server Options window.

★Advanced Options

The Advanced Options appear on a separate page if you checked the **Show advanced options** box on the Import Options page.

Create new copies of all jobs

If this option is checked, during the import adTempus will generate a new identity for each job in the import source. Use this option if you are importing jobs from another server and don't want the jobs to be updated if you import the same jobs again in the future.

Import group hierarchy under the following group

Use this option to import all jobs from the source to an alternate location in the job hierarchy.

Example

For example, suppose you have the following job group structure in the current instance of adTempus:

Root
- Group 1
- Group 2
- Server 2 Jobs

You are importing jobs created on "Server 2," which has the following group structure:

Root
- Group A
- Group B

If you check the **Import group hierarchy** option and select group "Server 2 Jobs" as the target group, the resulting hierarchy will be as follows:

Root
- Group 1
- Group 2
- Server 2 Jobs
  - Group A
  - Group B

Place imported jobs on hold

Use this option to place all imported jobs on hold so that you can review them before they begin executing. Each imported job's Hold Status will be set so the job is fully disabled (no execution allowed).

Next Step

Click **Next** to advance to the **Ready to Import** page or to the **Advanced Options** page and then to the **Ready to Import** page.

From the **Ready to Import** page, click **Next** to begin the import. When the import completes, the Wizard will advance to the Import Results page.

Related Concepts

Tools Reference 444
Data Import 446

Import Results Page

The **Import Results** page of the Import Wizard shows the status of each object you selected for import.

Each object will have one of the following statuses:

- **Saved.** The object was successfully imported and saved.
- **Ready to save.** The object was imported, but informational or warning messages were issued when adTempus tried to save the object. Review the messages by clicking the link in the **Messages** column. You can review and save the objects individually by clicking the **Edit...** button, or click **Finish** to save all "Ready to Save" objects.
- **Errors found.** The object could not be saved due to errors. You can review the error messages by clicking the link in the **Messages** column. You must then review and save the objects individually by clicking the **Edit...** button next to each object.

If you chose the **Save all imported objects if no problems are found** option on the Import Options page, adTempus has already saved the objects if possible. Otherwise, or if the save failed, you can review and save the objects.

If you chose the **Generate comparison report for existing objects** option, the **Differences** column will contain a link that you can click to view a comparison report for each object.

If you checked the **Save all imported objects** option or if you click the **Save All** button, adTempus attempts to save all of the imported objects in a single "transaction." If any object cannot be saved due to errors, then all updates are rolled back (no objects are saved).

If you edit a single object by clicking the **Edit** button and then save it by clicking **OK** in the properties window, only that object is saved.

If you click **Cancel**on this page, any objects with a status of "Saved" will remain in adTempus. Objects with a status of "Ready to save" or "Errors found" will be discarded.

Related Concepts

Tools Reference 444
Data Import 446

Job Templates

Job Templates

Job Templates allow you to create reusable templates (or "patterns") that you or other users can use as the basis for new jobs.

For example, suppose you have created a job that downloads a file from a Web site, reads the file using a script, and sends e-mail notification messages to various users based on the contents of the file. Several other users in your company need to setup up similar jobs, so you'd like to be able to share the work you've done with them.

You can do this by giving them a template based on your job. A template is similar to an export of the job, but with these important advantages:

- A Template can be used to create any number of new jobs based on the same pattern. On the other hand, once you import a job, if you import the job again it will update your existing job, rather than creating a new one.
- A Template can include Template Parameters, making it easy for other users to use the job. For example, your job downloads a file from a particular URL, but other people who use your template will need a different URL. You can replace the URL in your job with a Parameter such as `{URL to Download}`. When a user creates a new job based on your template, adTempus will prompt the user to enter a value for that Parameter and will insert it into the job. This allows jobs to be easily customized from the Template, without users needing to dig through the job changing settings.
- When you create a Template, adTempus removes credential information from the job, so you can share the template with others without worrying about user IDs and passwords being sent along with it.

Creating a Template

To create a template, first review and prepare your job, as described in the Preparing a Job to Be Templatized topic. Then use the Create Job Template Wizard to create the template.

Using a Template

To "install" a template in adTempus, copy the file that contains it to one of the following folders:

- To make the template available only to yourself, place the file in the "My Documents\adTempus\templates" folder.
- To make the template available to all users of a computer, place the file in the "templates" folder under the adTempus program folder (e.g., "c:\program files\Arcana Development\adTempus\templates\").

To use a Template, right-click the Jobs folder, a Job Group beneath that folder, or within the Job List and select the **New Job from Template** command that appears below the **New Job** command. If this command does not appear on the menu, close and restart the adTempus Console so it can detect your template.

adTempus will then present a list of the available templates. You will then be prompted for any Parameters used by the Template, and your job will be created and presented for review or editing.

Related Topics

Preparing a Job to Be Templatized 455
Create Job Template Wizard 456

Preparing a Job to Be Templatized

Before you create a Template based on a job, you may need to modify the job to make it easier to reuse.

Use Template Parameters

If there are values the user will need to change in the job after it is created, replace them with Template Parameters whenever possible. For example, if your job has a Web Request task that downloads a Web page, the template user will probably need to download a different URL than you do. Instead of leaving your URL in the job and requiring the user to change it, replace the URL with a Parameter. For example, if it the Web request's target URL was set to

```
http://www.example.com
```

change it to a parameter value such as

```
{URL to Fetch}
```

(be sure to include the curly braces). Then when you create the template using the Create Job Template Wizard, you will define the Parameter, setting the "Text to Replace" to "{URL to Fetch}" and the Prompting Text to something like "Enter the URL to download."

Now the user will be prompted for the URL when creating a job based on the template, and adTempus will plug the correct value into the job.

Use Job Variables

To make management of jobs as simple as possible, go beyond Template Parameters and use Job Variables for things such as file paths and URLs that the user may need to change in the future.

Extending the previous example, you can make the job easier to manage by replacing the URL with a Job Variable:

```
%URLToFetch%
```

Then create a new job-level Job Variable named "URLToFetch" and set its value to the Parameter value `{URL to Fetch}`.

Now when the user creates a job based on the template, adTempus will prompt for the URL and create a job where the "URLToFetch" Job Variable is set to the value entered by the user.

If the user needs to change the URL in the future, she can do so by simply modifying the value of the Job Variable, rather than needing to edit the step that executes the request.

Remove Notification Recipients

If your job sends notification messages, in most cases you don't want to deliver a Template that uses your notification recipients (unless the template user will be creating a job that notifies the same people). So before you create a Template, you should edit your job to remove the recipients from any Notification Actions, Notification Tasks, or File Capture Actions in the job.

Document the Job

Add brief comments to the job, steps, triggers, Responses, scripts, etc., to explain what the various pieces do. This will help the template user to understand how the job works and to make changes to it if required.

Related Concepts

Job Templates 454

Related Topics

Create Job Template Wizard 456

Create Job Template Wizard

The Create Job Template Wizard (**Tools > Import/Export > Create Job Template...**) helps you create a Job Template based on an existing job.

Before you begin creating a Template, review the Preparing a Job to Be Templatized topic and make any necessary changes to your job.

Select Job

Select the job you want to create a Template from. Each Template consists of a single job.

Template Parameters

If you want to define parameters for your template, add them here. For each parameter, you must enter the **Text to Replace**, which is the text that will be replaced with the parameter value entered by the user, and the **Prompt Text**, which is the message that will prompt the user for the parameter (e.g., "Trigger File Name"). You may optionally enter a **Description** of the parameter to give more detailed information about how it is used.

The **Text to Replace**must be enclosed in curly braces ({}), for example, `{MyParameter}`. Be sure to include the braces when you put the text in your job as well.

Before you begin creating the Template you must edit the job to add the parameters to it, as described in the Preparing a Job to Be Templatized topic.

Template Properties

Enter the basic information about your Template, which will be used to identify the Template to users.

You must enter a name and a brief description or summary of the template. Don't enter an extended description or usage instructions in the **Description**; there is a separate box for this information on the next page of the Wizard.

You may optionally specify the name of the template creator (your name and/or company name) to further help identify the template. If there is a Web page with more information about the template, you can include the URL.

Finally, you may enter a category for the template. The category is used to group templates with similar functions in the template selection window. For example, if your template creates a job to process Web pages, you might give it a category "Web Tasks."

Template Instructions

Here you may provide a detailed description of your template, instructions on how to use it, more information about how to enter parameters, etc.

You may enter plain text or text containing HTML tags for advanced formatting.

Save Template

Enter a file name and save your template.

By default the template will be saved in your private templates folder, and will only be available to you. To make it available to all adTempus users using this computer, save it in the "templates" folder under the adTempus program folder (e.g., "c:\program files\Arcana Development\adTempus\templates").

Related Concepts

Job Templates 454

Related Topics

Preparing a Job to Be Templatized 455

Arcana Scheduler Import

The Arcana Scheduler Import tool (**Tools > Import > Arcana Scheduler Import**) allows you to import scheduled programs from the Arcana Scheduler.

You will be able to review the imported jobs before they are saved.

Arcana Scheduler Client Components

To use the Arcana Scheduler import, you must have the Arcana Scheduler client components installed on the same computer as the adTempus Console. These components are not included in the adTempus setup, but are included in the Arcana Scheduler setup.

- If you already have the Arcana Scheduler installed on the computer where you are running the adTempus Console, the components are present.
- Otherwise, the Arcana Scheduler Import command will not appear on the Import menu, and you must install the components.

To install the client components, run the Arcana Scheduler setup. Select the "Custom" installation type, then choose only the "Arcana Scheduler ActiveX Component" feature.

Additional Considerations

- The default security settings for the Arcana Scheduler grant access only to members of the computer's Administrators group. To be identified as an Administrator, you must start the adTempus Console using the "Run as Administrator" option in Windows.
- You must be running version 2.3 or later of the Arcana Scheduler. If you are running an earlier version you will need to upgrade to version 2.3 and then run the import.
- adTempus cannot import the passwords associated with the user accounts on scheduled programs and network connections. Each time adTempus encounters a new user account during the import process it will prompt you for the password for that account.

Select Jobs

This page lists all Arcana Scheduler jobs that you have access to. Select the jobs you wish to import.

Click **Next** to set import options.

Import Options

The Import Options Page contains options that affect the import process.

Use the following set of holidays

Arcana Scheduler jobs that have the "Do not run on holidays" option checked will be configured to recognize the holiday set you specify. If you do not select a holiday set, the imported jobs will *not* be configured to skip holidays.

Disable imported jobs in the Arcana Scheduler

When this option is checked, adTempus will disable each job in the Arcana Scheduler after it is imported into adTempus.

If you do not select this option, be sure that you manually disable or delete the jobs, or disable or uninstall the Arcana Scheduler. Otherwise you will end up with the jobs being run by both schedulers.

Hold imported jobs in adTempus

Check this option to hold all imported jobs in adTempus. Otherwise jobs will be imported with the same status that they had in the Arcana Scheduler.

Regardless of whether you check this option, no imported job is saved until you have had a chance to review it.

Click **Next**to continue the import process.

Successful Imports

The Successful Imports Page lists all jobs that were imported successfully from the Arcana Scheduler.

To review an individual job, double-click the job's name in the list.

To save the jobs, select one or more jobs in the list and click **Save Selected Jobs**.

You must select and save jobs to save them in adTempus. Any jobs that you do not save on this page will be discarded when you leave it.

Failed Imports

The Failed Imports Page lists any jobs that had errors. To save these jobs in adTempus, you must review each job, fix the problems, and save it. To edit a job, double-click its name in the list.

Create Snapshot Window

Location: **Tools > Create Snapshot**

The **Create Snapshot** window allows authorized users to create an on-demand snapshot of all objects in adTempus. The snapshot can be used to revert to a previous version of an object, or to compare changes between snapshots.

The snapshot will appear in the Change Log for each object included in the snapshot. The snapshot will be retained for the number of days specified in the **Retain snapshots for** setting on the Auditing and Snapshots page of the General Server Options window.

You can also configure an automatic daily system snapshot of all data, or configure adTempus to create snapshots of individual objects automatically whenever objects are changed.

Related Concepts

Snapshot Overview 364
Auditing and Snapshots 364

Find and Replace

Location: **Tools > Find and Replace** or **CTRL+F**

The **Find and Replace** tool allows you to find jobs and other objects containing specified text, and optionally to replace text.

Other similar tools are available:

- To jump quickly to a particular job within the Console, use the Go to Job tool instead.
- To find all the places where an object such as a Credential Profile, Shared Script, Notification Recipient, etc., is used in adTempus, use the **Find/Replace References** command available on the pop-up menu for the object. For example, right-click a recipient in the Notification Recipients view, then select **Find/Replace References**from the pop-up menu. To find or replace user credentials, right-click the profile in the Credential Profiles window and use **Find/Replace References**. This command will open the Object References window.

This tool does not search for the names of shared objects. For example, you cannot use it to find all the places where you use a particular Shared Script, Notification Recipient, Credential Profile, etc. To locate references to those objects, use the Find/Replace References tool instead.

After you configure the search options, click the Find button to begin the search/replace operation (see below).

Search Options

Search Jobs

Search Jobs

Check this option to search within jobs. Select the groups or jobs in which you want to search.

Search job history

When this option is checked, adTempus will search the history of the selected jobs in addition to the job definitions. Depending on the amount of history present and the options you select, this may be a slow process.

Log messages

adTempus will search the text of all messages in the Job Log for the selected job(s).

Captured files

adTempus will search the file names of all Captured Files for all instances of the selected job(s).

Contents of captured files

adTempus will search within the Captured Files for all instances of the selected job(s). For example, this option could be used to search for a particular error code within a log file captured by the job.

When you select this option, specify the patterns for the files that adTempus should search. Captured Files whose names do not match any pattern are ignored.

You can also exclude certain files by prefixing a pattern with -. For example, to include all files with a "txt" extension except captured console output, you could use this pattern:

```
*.txt; -Captured Console Output.txt
```

Search Elsewhere

Options on this page allow you to search within objects other than jobs. For example, you can search for a particular string within the code of a Shared Script.

If you are searching jobs, the search automatically includes any objects referenced by the jobs. For example, if adTempus is searching in a job that executes a Shared Script, that Shared Script will be searched even if you don't select the Shared Scripts option here. Use the options on this page to search in objects that may not be referenced by the jobs you are searching, or when you want to search in those objects without searching any jobs.

Search For

Specify the text you want to search for. You can use Regular Expressions for more flexibility.

Optionally you can replace the found text.

If you are using Regular Expressions, the replacement text can contain references to capture groups using the syntax "$1" (first capture group), etc.

Search and Replace

Search

Click **Find** to begin the search process. adTempus will list all objects that match the search text, along with information about where the search text was found.

From the results list you can view or edit the objects where the text was found.

If you checked the **Replace** option, you will see the replacement values when you view or edit the objects, and if you click **OK** while editing an object, it will be saved with the new values.

Replace

To replace text, select the objects that you want the replacement applied to and click **Save Selected**. You may optionally choose to have adTempus create Snapshots of all affected objects before the changes are applied, allowing you to revert to the previous version if necessary.

When you click **Save Selected**, adTempus saves the objects in a single transaction; if any object fails to save, then all changes are rolled back.

Related Topics

Object References Window 443

Go to Job

The **Go to Job** tool allows you to jump quickly to a particular job within the adTempus Console.

This tool is especially useful if your jobs are organized into many Job Groups, and you cannot remember which group a particular job is in. Rather than looking through all of the Groups in the Console, you can simply use the Go to Job tool to jump straight to the job, whatever group it is in.

To start the Go to Job tool, select **Go to Job** from the **Tools** menu or simply press `CTRL-G`). This command can be used from any folder in the Console, and will search all jobs within the currently-selected server.

⭐In the **Go to Job** window, type the name (or part of the name) of the job you want to find. You can type the beginning of the name or text that appears anywhere in the name. As you type, the job name box will suggest jobs whose name starts with the characters you have typed.

The **Matching jobs** list will fill with the names of all jobs that contain the text you have typed. To jump to a listed job, double-click the job's name, or select the name and click OK, or simply press **`Enter`** to jump to the highlighted job.

The Console will then jump to the selected job in the Console, opening the containing Job Group if necessary.

To search for a job based on text within the job (such as a file name or text in the job description), use the Find and Replace tool.

Report Designer

Report Designer

Location: **Tools > Report Designer**

This feature is new for adTempus version 4.

The Report Designer tool allows you to customize the reports included with adTempus or to create your own reports.

Report Designer reference

Select the **Report Designer Reference** item from the Help menu in the Report Designer for a reference guide to using the Report Designer.

Please check the online version of this topic for the latest information about designing custom reports, or visit the adTempus Reporting Forum.

Additional documentation on designing custom reports will be available soon. Please see the adTempus Reporting Forum for assistance with creating custom reports in the meantime.

Related Concepts

Reports 383

Open Report Window

Select the report that you want to modify. You can choose a loaded report (a report that is listed in the **Reports** folder of the Console Tree) or open a report file previously saved from the report designer.

Related Concepts

Report Designer 463

Create Report Window

To create a new report you can start with an existing report, or start from scratch using one of the data sources listed.

Related Concepts

Report Designer 463

Report Properties Window

The **Report Properties** window appears when you save a report.

Report Title

Provide a name for the report. This is the name under which it will be listed in the Console.

Description

Provide a description of what the report does.

Creator

Optionally enter the name or company name of the creator of the report.

Report ID

Enter a unique report ID, which will be used to identify the report internally in adTempus. A default Report ID will be constructed for you from the **Creator** and **Report Title**.

Related Concepts

Report Designer 463

Job Status Descriptions

The following table describes the statuses that each job or step may display.

Status	Meaning
Abandoned	The adTempus service was shut down while the job was running, so adTempus does not know the status of this job or step. The program launched by adTempus may still be running.
Aborted	The job was terminated by a user.
Condition Failed	An error occurred while evaluating one or more conditions for the job or step. Check the Job Log for more information.
Condition Not Met	One or more conditions for the job or step were not met.

Status	Meaning
Dispatched to Agent	The job has been dispatched to the Remote Agent and the Master is waiting for a status update from the Agent.
Execution Failed	The task could not be started.
Failed	The job or step failed. Check the Job Log for more information.
Killed	The step was terminated by adTempus because it exceeded the maximum execution time specified in the step's properties.
Missed	The instance was missed because adTempus was not running.
Not Run	The job or step was not executed.
Queued	The job has been triggered and is waiting to be executed.
Resource Failed	One or more resources for the job could not be loaded.
Resubmitted	The job or step was restarted as a result of a Job Control Action.
Running	The job or step is running.
Skipped	The job or step was skipped. For a job, this will occur if another instance is already running, and the job is configured to not start a second instance. For a step, this will occur if the Program Execution Task is configured to not execute if the program is already running.
Starting	The job or step is starting.
Succeeded	The job or step completed successfully.
Trigger Failed	An error occurred in one or more of the job's triggers. Check the Job Log for more information.
Waiting for Agent	The job is waiting to be dispatched to a Remote Agent.
Waiting for Condition	The job or step is waiting for a condition to be met.
Waiting for Previous Instance	The job is waiting for a previous instance of the job to complete.
Waiting for Resource	The job is waiting for a resource to be available.
Waiting for Restart	The job or step is being restarted as a result of a Job Control Action, but the wait time has not yet elapsed.
Waiting for Trigger	The job is waiting to be triggered.

The adTempus Service

The Service

The adTempus service is the "engine" that does the work of executing adTempus jobs. The adTempus service must be installed on each computer on which jobs are to be executed.

If you have configured more than one instance of adTempus on a computer, there will be a service for each instance.

Managing the adTempus Service

The adTempus service can be started or stopped using the Windows **Services** tool (found in the **Administrative Tools**folder on the Start menu). Once the service is started, all management of adTempus is done using the adTempus Console.

Troubleshooting Service Problems

If the adTempus service cannot be started, the Application log in the Windows Event Viewer will contain one or more error messages indicating what the problem is. See the Service Startup Troubleshooting topic for more information.

Service Account Requirements

For adTempus to be fully functional, the adTempus service must be configured to run under the Local System account, as this is the only context in which adTempus can fully integrate with Windows.

If you are using a remote database server to host the adTempus data, this restrictions means that you must either use SQL Server security, or grant SQL Server access to the machine account of the computer where adTempus is running (see the Database Credentials topic for more information).

If absolutely necessary, it is possible to run the service under a user account instead, but some adTempus features will not be available (see below).

The account that you use for the adTempus service must meet the following requirements:

- The account must be a member of the Administrators group on the computer.
- The account must have the following rights granted through the Local Security Policy:
  - Log on as a service
  - Act as part of the operating system
  - Create a token object
  - Replace a process level token
- The account must be granted read, write, and backup permissions on the database server.

Before you can start adTempus using a user account for the service, you must set a Registry option to enable this configuration:

- Run the Registry Editor and go to the key **`HKEY_LOCAL_MACHINE\Software\Arcana Development\adTempus\Instances\Default\Options`**. (Replace "Default" with the instance name if you are configuring an adTempus instance other than the default.)
- Create a new DWORD value named `Server:AllowServiceAsUser` and set it to `1`.

Limitations When Running the Service Under a User Account

When the adTempus service is run under a user account instead of the Local System account, interactive program execution will be disabled (it will not be possible for a job to execute a program that is visible to the user). The only option available for the User Interaction setting for jobs will be "Hidden," and the **Make the job visible on my desktop** option will not be available in the Execute Job window when you run a job manually.

Machine ID and Database Ownership

When you start the adTempus service for the first time on a new computer, adTempus generates a **Machine ID** to uniquely identify this instance of adTempus. This value is stored in the Registry and also in the adTempus database.

Database Ownership

If another adTempus instance (running on the same computer or a different computer) tries to connect to the adTempus database, the service will fail to start and will log the following message to the Application log in the Event Viewer:

> The adTempus database belongs to another instance of adTempus. To switch ownership to this instance, start the adTempus service with the "resetdbowner" option.

This is done to protect against accidentally running two adTempus instances against the same database. Depending on your scenario, use one of the following options to override this error condition:

- The adTempus software has been transferred to a new computer and the new installation should now be the owner of the database: Use the "resetdbowner" startup option to permanently reset the ownership information in the database.
- You have made a copy of the adTempus database for use by the adTempus software on a new computer: Use the "resetdbowner" startup option to permanently reset the ownership information in the database.
- The adTempus service needs to use this database temporarily (e.g., for testing or disaster recovery failover). Use the "ignoredbowner" startup option to temporarily ignore the database owner. Before doing this, ensure that no other adTempus instances are running using this database.

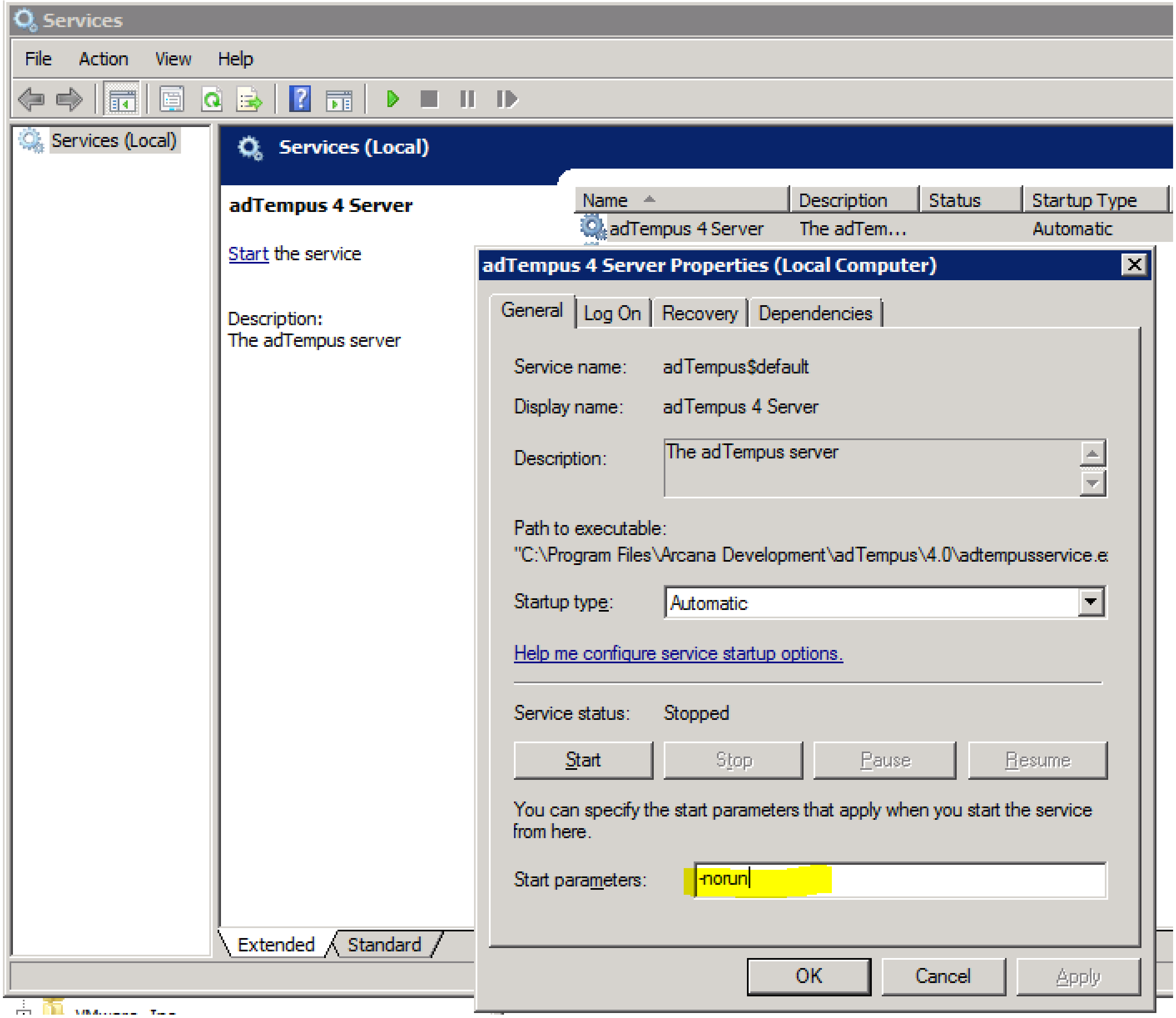
Resetting the Machine ID

If you have cloned the server where the adTempus service is installed, you should also reset the Machine ID on the cloned server if adTempus is still running on the new server. To do this, restart the adTempus service using the "resetidentity" startup option.

Service Startup Options

Various commands can be sent to the adTempus service through the use of service startup options. These options are sent to adTempus using the **Start parameters** box when starting the adTempus service using the Windows **Services** tool.

To apply these options, stop the service, then enter the appropriate option(s) in the **Start parameters** box and start the service.



adTempus 4 Server

Start the service

Description:
The adTempus server

Service name: adTempus$default

Display name: adTempus 4 Server

Description: The adTempus server

Path to executable:
"C:\Program Files\Arcana Development\adTempus\4.0\adtempusservice.e

Startup type: Automatic

Service status: Stopped

You can specify the start parameters that apply when you start the service from here.

Start parameters: -norun

Options set this way only apply until the next time the adTempus service is restarted.

Available Options

The following options are available:

Option	Description
-norun	Disables all job execution until the service is restarted without the option. Note: This option is only available beginning with adTempus 4.0.3. For earlier versions, use one of the other available methods for suppressing

Option	Description
	job execution.
-nostartup	Suppresses all Startup Triggers. All other triggers work normally.
-purgemirroreddata	When applied to an Agent, causes the Agent to purge all jobs and other data that it has received from the Master.
-resetmaster	When applied to an Agent, causes the Agent to discard its attachment to its Master server. Once this is done, it will attach to the first Master that connects to it. When you use this option, all data from the current Master is discarded (the same as if the "-purgemirroreddata" option is set). The Master can also be reset using the Engine Mode utility.
-resetidentity	Carries out the same actions as "-resetmachineid" and also regenerates the encryption keys used for encrypting and signing communications with other adTempus servers.
-resetmachineid	Resets the Machine ID that identifies this adTempus instance. Using this option also applies "-resetdbowner". See Machine ID and Database Ownership.
-resetdbowner	Resets the owner information in the adTempus database. See Machine ID and Database Ownership.
-ignoredbowner	Temporarily ignores the owner information in the adTempus database, allowing adTempus to start using a database that belongs to another instance. See Machine ID and Database Ownership.

Utility Programs

Server Tools Launcher

The Server Tools Launcher provides access to the following tools, which can also be started using the associated shortcuts in the adTempus group on the Start menu:

- Engine Mode Utility
- Administrator Provisioning Utility
- Database Utility
- Database Installation and Configuration Wizard
- License Manager
- Endpoint Configuration Tool

adtChkpt Utility

The adtChkpt utility can be used to set or retrieve the checkpoint for an executing job. This utility is meant to be used when you have a batch file that executes several different processes and you want to let adTempus know where things stand in case the job needs to be restarted from this point later.

This program relies on information put in the environment by adTempus and therefore will only work within a batch file run by adTempus.

To set the checkpoint, use the syntax

```
adtchkpt /s "checkpoint name"
```

Where "checkpoint name" is whatever string you want to pass as the checkpoint.

See the Checkpoints topic for an example of how to use checkpoints in a batch file.

Command-Line Job Execution Utility (adtExec)

The Command-Line Job Execution Utility can be used to execute or terminate adTempus jobs from the command line, batch files, scripts, etc. You can use this utility, for example, to start a job from a batch file, or to create a desktop shortcut that starts a job.

The program is named "adtexec.exe" and can be found in the adTempus program directory. This tool is command-line driven and does not present an interactive user interface.

adtExec uses the same authentication as the adTempus Console: the user executing the utility must have Execute permission for the job.

Command-Line Syntax

adtExec has the following syntax:

```
adtexec job [options]
```

All parameters except the job are optional.

Parameter	Meaning
job	Specify the name (enclose in quotes if it contains spaces or other punctuation) or Job ID (available from the job properties window) of the job you want to execute. If the job name is unique across all Job Groups, you do not need to include the group name. However, if you have more than one job with the same name you must include the group name (see Examples below).
-acknowledge	When used in conjunction with the -terminate parameter, this option marks the aborted instances as "Acknowledged" so they do not appear in the Failed Jobs view in the Console. You must also specify the -wait parameter when using this option.
-agents:"agentName" [...]	Runs the job on the specified agent(s) only. Specify multiple values as show or by repeating the -agents parameter on the command line.
-checkpoint:"checkpoint"	The checkpoint value to pass to the job
-force[{+|-}]	Forces a new instance of the job even if an instance is already running and the job's settings specify otherwise. Default value: True.
-forcemaster[{+|-}]	Forces the job to run on the Master even if it is not configured to run there. Default value: False.
-ignorejobconditions [{+|-}]	Ignores job-level conditions. Default value: True.
-ignorestepconditions [{+|-}]	Ignores step-level conditions. Default value: False.
-instance:instancenumber	When used in conjunction with the -terminate parameter, specifies the instance to terminate.
-list	Lists all jobs that you have permission to execute; does not execute jobs. If you specify a partial job name, all jobs matching that name will be listed. For example `adtexec warehouse -list` will list all jobs containing the text "warehouse".
-masteronly[{+|-}]	Run the job only on the Master even if it is configured to run on Agents. Default value: False.
-user:"userID"	The Windows user ID to use when connecting to adTempus. If not

Parameter	Meaning
	specified, the connection is made under the identity of the account running adtexec.
-password:"password"	The password to use when connecting to adTempus. If not specified, the connection is made under the identity of the account running adtexec.
-responses:value	Determines which Responses will be executed for the job. Valid values: • all: Execute all Responses • none: Do not execute any Responses • nojob: Execute Responses except Job Control actions Default value: all.
-server:"serverName"	The name or IP address of the adTempus server to connect to. To connect to the local server, do not specify a value.
-steps:"stepNumber"	Specify the step number(s) to execute. You may use ranges or lists. For example: "2-4", "3-", "1,3,4"
-terminate	Terminates the specified job. If the -instance parameter is given, only that instance is terminated; otherwise all active instances are terminated. Use -wait to wait for all/specified instances to end, otherwise the tool returns immediately. Use -acknowledge to acknowledge the status of the terminates instance(s) so they do not appear in the Failed Jobs view.
-variables:"name=value" [...]	Overrides for job variables defined for the job. Specify multiple values as show above or by repeating the -variables parameter on the command line.
-wait[{+|-}]	Wait until the job finishes executing. If not specified, adtexec returns immediately after submitting the job. Default value: False.

Examples

The following command runs job "myjob" in group "group1" and waits for it to finish, using the following settings:

- Only steps 2-4 are executed.
- Job Control Responses are not executed.
- The job is not started if another instance is already running.
- The "ExecutionSource" job variable is set to the value "CommandLine".

```
adtexec "group1\myjob" -wait -steps:"2-4" -responses:nojob -force- -variables:"Execu
```

adTempus Database Utility

The adTempus Database Utility provides a means of interacting directly with the adTempus database.

To start the program, run the **Database Utility** from in the adTempus Server Tools group on the Windows **Start** menu. The program will connect automatically to the adTempus database. Once the program has connected to the database, the following operations are available:

- Backing up the database
- Restoring the database
- Querying and updating the database

Never use this tool (or any other) to update the adTempus database except under the direction of support personnel. Doing so may result in irreversible corruption of your data.

Backing Up the Database

The Backup command (**Tools > Backup Database**) is used to create a complete backup of your adTempus database, which can later be restored using the Restore command.

You can configure adTempus to create daily automatic backups.

This tool is intended for use with the SQL Server Express database engine installed with adTempus. If you are using your own database server, you should use the backup and restore tools provided in the server's management software instead. If you use this tool with a remote database server, note that the file path you enter for the backup file will be evaluated on the database server, and must therefore be valid on that computer, not on the computer where you are running the tool.

This topic covers database backups, which are used to recover your entire adTempus database in the event of a failure. Database backups cannot be used to restore selected jobs or other objects. Configure automatic snapshots to allow you to recover data at a more granular level.

Restoring the Database

The Restore command (**Tools > Restore Database**) is used to restore your adTempus database using a backup file created by the Backup command or by an automatic backup.

This topic covers database backups, which are used to recover your entire adTempus database in the event of a failure. Database backups cannot be used to restore selected jobs or other objects. Configure automatic snapshots to allow you to recover data at a more granular level.

This tool is intended for use with the SQL Server Express database engine installed with adTempus. If you are using your own database server, you should use the backup and restore tools provided in the server's management software instead. If you use this tool with a remote database server, note that the file path you enter for the backup file will be evaluated on the database server, and must therefore be valid on that computer, not on the computer where you are running the tool.

The Restore operation will overwrite the adTempus database with the backup; you will lose any changes made since the backup was taken.

Querying and Updating the Database

The main screen of the Database Utility presents an empty box in which you can enter a SQL query or update statement.

Press **F5** to execute each statement.

Never use this tool (or any other) to update the adTempus database except under the direction of support personnel. Doing so may result in irreversible corruption of your data.

Engine Mode Utility

The adTempus Engine Mode utility (adtenginemode.exe) is found in the adTempus program files folder and in the **adTempus Server Tools** group on the Windows **Start** menu. It is used to change the Engine Mode in which the adTempus service on a computer runs:

- Standalone
- Master
- Agent / Standalone
- Agent Only

See the adTempus Engine Mode topic for information on these modes.

Resetting an Agent's Master

An Agent can belong to only one Master. The Agent will "lock on" to the first Master that connects to it, and reject any subsequent connection attempts from other Master computers.

To move an Agent to a new Master, follow these procedures:

1. In the current Master, delete the Remote Agent definition for the Agent (in the Remote Agents folder in the Console). This is necessary so the current Master does not try to reconnect to the Agent.
2. On the Agent computer, run the Engine Mode utility.
3. Check the **Reset Master** box.
4. Click **OK** to save the changes, and **OK** again to allow the tool to restart the adTempus service.
5. On the new Master, create a Remote Agent definition for the Agent (in the Remote Agents folder in the Console). The new Master will now connect to the Agent, and the Agent will lock on to the new Master.

When you reset the Master for an Agent, the Agent will discard all jobs and other data that were sent from the previous Master.

Endpoint Configuration Tool

The Endpoint Configuration Tool can be started from the adTempus Server Tools group on the Windows **Start** menu, or from the Server Tools Launcher.

This tool allows you to select the server certificate that the adTempus server will use to secure its communication channels.

Create Certificate

The **Create Certificate** button creates a new, self-signed server certificate and selects it for use by the server.

Technical Information

Data Security

All adTempus data is stored in the adTempus database, except captured files, which are stored in the "data\capturedfiles" subdirectory under the adTempus program directory.

The adTempus database and the captured files should be secured against unauthorized access using Windows file system security. For adTempus to operate, the SYSTEM account must have full control over the "data" subdirectory (and its contents and subdirectories). adTempus users do *not* need access to the data files.

User IDs and passwords within the adTempus database are stored in an encrypted format to protect against users who might gain access to the database. They are also encrypted when exported to a file.

All data is encrypted before being transmitted between the adTempus Console and Service, so it is protected when transmitted over a network connection.

Server Security Certificates

adTempus uses TCP/IP for communication between computers. This includes communication between the Console and the adTempus server, between two adTempus servers that are linked, and between Master and Agent servers when you are using Distributed Scheduling.

If the computers are both part of the same Windows security domain or there is a trust relationship between the computers or their domains, the communication channel is secured by Windows.

Otherwise, you must install an X.509 server certificate on the adTempus server.

Obtaining a Server Certificate

If the server where adTempus is running already has a commercially-issued certificate, that certificate can be used for adTempus as well.

Otherwise you will need to obtain a certificate through one of these means:

Commercially-Issued SSL Certificate

You can obtain an SSL certificate (which is an X.509 certificate) from a commercial issuer. However, such certificates can be costly and a commercially-issued certificate is not necessary for adTempus.

Internally-Issued Certificate

You can obtain a certificate from your domain's or organization's certificate authority. Contact your security administrator for assistance with obtaining and installing a certificate.

Note that when you use this approach, the root certification authority used for the certificate must be trusted by all computers that will be connecting to adTempus. You therefore may need to install the authority as an additional trusted authority on those computers or domains.

Self-Signed Certificate

For testing or low-risk situations you can use a self-signed certificate, which is a certificate issued locally on the computer, rather than by a certificate authority. You can generate a self-signed certificate using the Endpoint Configuration Tool. Note that a self-signed certificate will not be trusted by other computers, and you will be prompted to review and trust the certificate when you connect to the server.

Configuring adTempus to Use the Certificate

Once a suitable certificate has been installed on the server, run the Endpoint Configuration Tool to configure adTempus to use the certificate.

Backing Up and Restoring Data

To reduce the chances that adTempus configuration or history information is lost in the event of a hardware or software failure, it is important to back up your adTempus data frequently.

This topic covers database backups, which are used to recover your entire adTempus database in the event of a failure. Database backups cannot be used to restore selected jobs or other objects. Configure automatic snapshots to allow you to recover data at a more granular level.

Backing Up Data

adTempus data resides in two locations:

- Captured files are stored in the "CapturedFiles" folder under the adTempus instance directory (e.g., "c:\program files\Arcana Development\adTempus\instances\default\data\CapturedFiles"). If captured file data is important, be sure your backup routine includes this folder.
- All other adTempus data (job definitions, history, etc.) is stored in the adTempus database, which should be backed up regularly as described below.

If you are using the default SQL Server Express database engine installed with adTempus, adTempus automatically backs up your database each day at about 1:30 AM (backup settings are available in the Server Options window). The database backup file is saved in the "backup" folder under the adTempus instance directory (e.g., "c:\program files\Arcana Development\adTempus\instances\default\data\database\backup"). You should ensure that your backup routine includes this folder.

If you are using your own database server to host the adTempus database, adTempus will not automatically create backups by default. Therefore you must make sure that your existing database server backup routine includes the adTempus database, or configure adTempus to perform the backup.

In addition to automated backups, you can use the adTempus Database Utility to manually back up the database at any time.

Restoring Data

Restoring an adTempus installation involves restoring both the adTempus database and the CapturedFiles directory. (If you do not restore the CapturedFiles adTempus will operate normally, but you will receive an error if you try to view a captured file.)

Note that restoring the adTempus database restores your configuration and history to the state they were in at the time the backup was taken; any subsequent changes are lost.

Before restoring the adTempus database and files, stop the adTempus service.

If the backup was performed on your own database server, you should use the tools on that server to restore the database.

If the backup was produced automatically by adTempus, or using the adTempus Database Utility, use that tool to restore the database.

Port Information

adTempus uses the TCP protocol for communication between the client and the server. The server listens on port 3760; the client uses a dynamically-assigned port.

If you are using remote administration and your server is behind a firewall, you will need to make sure that the firewall is configured to allow incoming connections on port 3760.

Changing the Port

If for some reason you need adTempus to use a different port, you can do so as follows:

1. Using the Registry Editor, locate the key `HKEY_LOCAL_MACHINE\Software\Arcana Development\adTempus\Instances\Default`. (If you are changing the setting for an adTempus instance other than the default instance, replace "Default" with the instance name.)

2. If this key does not have a subkey named `Options`, create one.

3. Under the Options key, create a new **string** (REG_SZ) value called `Server:Port`. Set this value to the port number you want to use. Important: the registry value must be a string value, not a DWORD value.

4. You must restart the adTempus service for the change to take effect.

Users must specify the correct port number in the Server Connection window when connecting to the server. If the server is connected to as a Remote Agent or a Linked Server, the port must be specified in the Remote Agent properties or Linked Server properties when defining the connection.

Notes

Exit Codes

adTempus does not know what, if anything, a particular exit code from your program means. It only reports the information it receives from the program, and responds according to the rules you have defined.

The meaning—if any—is determined by whoever wrote the program.

Whenever a program runs, it passes a numeric exit code (or return code) back to the operating system (or in this case adTempus) to indicate its status.

While all programs return an exit code, not all programs return a meaningful exit code.

By convention, programs return a value of 0 to indicate successful completion or some value greater than 0 to indicate a warning or error condition.

Generally, programs that were designed to be run in a batch fashion will return some sort of useful exit code. This value can be used by adTempus to determine the success or failure of the task (see the success criteria for the Program Execution task).

adTempus does not know what the exit codes from a particular program mean, if anything. Check the documentation for the application in question, or check with its maker.

Remember, it's entirely possible that the exit code is meaningless. adTempus defaults to treating anything other than 0 as a failure, because that's the convention. If you program appears to be working properly but keeps sending back an exit code of 42, you can change the success criteria for the step so that adTempus won't fail it.

If your program does not return a meaningful exit code, you can configure adTempus to ignore the exit code, or use the Success Rule Script to determine whether the program succeeded by reading an output file from the program or checking some other indicator.

How to Set the Exit Code

How you set the exit code depends on what development language you're setting it from.

C/C++ Programs

The exit code is the return value from the main() or WinMain function.

.NET

If you are using VB.NET or C#, you need a main function that returns a value; that's the exit code. See the .NET documentation for details.

Visual Basic

If you are using Visual Basic .NET, you need a main function that returns a value; that's the exit code. See the VB.NET documentation for details.

If you are using an earlier version of VB, there is no supported method for returning an exit code. Microsoft used to recommend the approach described below. We give it to you here for your convenience, but please be aware that they no longer recommend or support this approach, due to the problems described in their knowledge base article Q288216. Show me how.

This approach uses the Windows API ExitProcess function to terminate the VB program, returning an exit code. Please read this before you use this method.

```
Private Declare Sub ExitProcess Lib "kernel32" (ByVal
exitCode As Long)
ExitProcess(42)
```

Replacing, of course, 42 with whatever exit code you want to return.

In addition to the problems noted by Microsoft, we'll point out that calling the ExitProcess function while your program is being debugged in the Visual Basic development environment will cause the Visual Basic IDE itself to terminate, along with your program.

A better approach would be to a) use something other than VB, or use a newer version of VB or b) use a file to signal success. You can then use a script to check for the presence of this file and set the step's status accordingly.

Batch Files

The exit code from a batch file is the exit code of the last process the batch file executed. If you want to set the exit code of your batch file to something in particular, use the exit command:

```
exit /b 42
```

Sets the exit code to 42.

Scripts

If you're using a Script Execution task to run the script you can return an exit code as described here. If you're executing the script in some other fashion, you can use the WScript.Quit method to return an exit code.

Everything Else

Check the documentation for whatever development language you're using.

Related Topics

Failed Jobs 369
Troubleshooting Failed Jobs 370

How Terminates a Process

Due to issues with security and Windows session isolation, adTempus is generally not able to close a program by sending a "Close" command to its main window and allowing it to close gracefully. Instead, it kills the process, equivalent to using the **End Process** command in the **Processes** list in Windows Task Manager.

When a program is terminated in this fashion, it is killed immediately and does not have a chance to do any of the things it would normally do before exiting (things like saving files).

Programs terminated in this manner generally have an exit code of 255.

Network Access for Jobs

If your jobs need access to files on network devices, please review the following information.

Mapped versus Unmapped Network Drives

There are two approaches applications typically use for referring to network resources.

UNC Paths

The preferred approach is to refer to the resource using its UNC designation. If your program needs to read the file `data1.dat` on share `proddata` on server `prod1`, it would refer to the file as `\\prod1\proddata\data1.dat`.

Drive Letters

Unfortunately some applications—especially those that were not originally developed with batch execution in mind—require drive letter mappings instead. In this approach, the application relies on a particular drive letter being mapped to the network resource.

For example, using our resource from the previous example, the drive letter K: is mapped to `\\prod1\proddata\`. The application then refers to the file as `K:\data1.dat`.

Drive letters may be mapped using Windows Explorer and through batch files (using the `net use` command).

Using UNC Paths

If your application (program, batch file, script, etc.) refers to network resource using UNC paths, you generally do not need to take any special steps to provide access. The one exception to this is if you need to use different credentials for the network resource than you are using for the job. In this case, you will need to add a network resource to the job, connecting using the appropriate credentials.

Using Drive Letters

If your application uses mapped drive letters, we strongly recommend changing it to use UNC paths instead, so that it does not have to rely on drive letters being correctly mapped. If this is not possible, you will need to set up drive letter mappings in adTempus, as described in the remainder of this section.

Drive letter mappings required by your application **must** be mapped on the Resources page of the job's properties. You must not rely on drive mappings that were made outside of adTempus.

For example, you may have used Windows Explorer to map the drive letter K: to a particular network resource. When you run your program yourself, it is able to read data from drive K: without any problems. However, this drive letter mapping belongs to your logon session, and is only available to programs that run in the same logon session. If the job runs under a different user account, it will not have access to this drive mapping. Further, even if the job runs under your user account, this drive mapping is only loaded if you are logged in interactively at the computer. If the job runs while you are not logged in interactively, the drive mapping will not exist.

Related Topics

Network Resource 416

Regular Expression Syntax

A *regular expression* defines a pattern to be matched. A regular expression can be a simple string (for example, if you want to find the word *unhandled*, you would use the regular expression `unhandled`.

Regular expressions can be used to match far more complex patterns, though. For example, you may want to match error messages that contain particular error codes. For example, a particular program may issue error messages formatted as "Error code: XX12345E", where `XX12345` is the error code and the `E` suffix indicates a severe error. You want to find all error messages with this severe error suffix. The regular expression to do this would be:

```
Error code: .{7}E
```

This pattern would match any message containing "Error code: " followed by any seven characters and an *E*.

The following sections provide an overview of the syntax of regular expressions.

adTempus uses the Microsoft .NET regular expression syntax. Matching is always case-insensitive.

General Rules

Regular expressions use the following special characters:

. [ ] - ^ * ? + $ | ( ) { } \

Any characters other than these special characters match themselves. For example, `shakespeare` matches `shakespeare`.

To match a special character, you must precede the special character with a backslash (\). For example, if you want to match a string that contains the sentence "Continue with deletion?", you cannot use `Continue with deletion?` as your regular expression, because the question mark has a special meaning in regular expressions. Instead, you would have to use `Continue with deletion\?`. The backslash indicates that the following character should be treated as a plain character, not a special character.

Regular expressions used in adTempus are not case sensitive.

Single-Character Pattern Matching

The . (period) character matches any single character. For example, `c.r` would match `car` or `cur`.

Lists

The [and] characters (left and right square brackets) are used to denote a list of valid characters. For example, `d[ou]g` would match `dog` or `dug` but not `dig`.

The - character is used within brackets to specify a range of valid values. For example, `d[a-z]g` would match any string that began with d, ended with g, and had any letter between a and z (inclusive) in between.

The ^ character is used within brackets to specify characters that should not be matched. For example, `d[^oi]g` would match any three-character sequence beginning with d and ending with g except `dog` and `dig`.

Repetition

The * (asterisk) character is used to match zero or more occurrences of the previous regular expression. For example, `b[a-z]*e` would match the letters *b* and *e*, with zero or more letters in between (`be`, `bare`, `bore`, etc.).

The + (plus) character is the same as *, but requires at least one occurrence of the pattern. For example, the expression b[a-z]+e would match the letters *b* and *e*, with one or more letters in between (`bare`, `bore`, etc.).

The ? character indicates that the preceding regular expression is optional. For example `ca?r` would match *cr* or *car* but nothing else.

The {} characters can be used to specify repetition of the preceding regular expression. The repetition can be specified using one of the following syntaxes:

{count}
: matches *count* occurrences of the preceding expression

{min,}
: matches *min* or more occurrences of the preceding expression

{min,max}
: matches at least *min* but no more than *max* occurrences of the preceding expression

Grouping

Parentheses () can be used to group characters together to delimit subexpressions. For example, the regular expression `X12+` matches "X1" followed by one or more 2s. `X(12)+`, on the other hand, matches "X" followed by one or more "12"s.

Alternation

The | (vertical bar) character denotes alternatives between regular expressions. When placed between two regular expressions, the alternation character means that the pattern will match any string that matches either of the expressions. For example, `(Error|Warning) Message` would match the strings "Error Message" or "Warning Message".

Predefined Variables

When adTempus runs a job, it sets several job variables automatically. These variables can be used within the job by using substitution tokens. For example, you can include the following text in the command-line parameters for a program to pass it the name of the job: `%ADTJobDescription%`.

These variables are also available to scripts, and are defined as environment variables for any programs run by the job.

When you are editing a text field that supports Job Variables, you can use the Insert Variables button ( ) to open a text edit window that lets you selected the predefined variables from a list.

Miscellaneous

Miscellaneous variables

Variable	Description
NotificationFromName	The display name to use as the sender for notification messages. Overrides setting in SMTP server properties.
NotificationFromAddress	The address to use as the sender for notification messages. Overrides setting in SMTP server properties.
DefaultMessage	The default notification message generated for a Notification Action.
DefaultSubject	The default notification subject generated for a Notification Action.
Message	The current notification message for a notification operation.
MessageTimestamp	The timestamp for a notification message.
ComputerName	The name of the computer on which adTempus is running.

Variable	Description
ServerName	The name of the computer on which adTempus is running.
adTempus.InstanceName	The adTempus instance name.
adTempus.ServerAndInstanceName	The name of the server and instance, e.g., "computername\instancename". If this is the default instance, will be only "computername".
ADTServerPath	The path where adTempus is installed.

Job

Variables defined for all jobs

Variable	Description
ADTJobID	The unique identifier for the job.
JobStatus	The current status of the job.
ADTFullyQualifiedJobName	The fully-qualified name of the job, including the group name.
ADTJobName	The name of the job, without the group name.
ADTTriggerClass	The type of trigger that started the job.
ADTQueueName	The name of the Queue to which the job is assigned.
ADTGroupName	The full name of the Group to which the job is assigned.
ADTExecutionReason	Text description of the reason the job is being executed.
ADTJobRestartCount	The restart attempt number. Will be 0 for first execution attempt, then incremented each time the job is restarted by a Restart action.
ADTExecutionAttempt	The execution attempt number. Will be 1 for first execution attempt, then incremented each time the job is restarted by a Restart action.
ADTJobNotes	The Description/Notes information for the job.
ADJobInstance	The OID of the ExecutionHistoryItem representing this instance of the job.
ADTInstanceID	The instance number for this instance of the job.
ADTJobTemp	A temporary directory that is automatically deleted at the end of the job.
ADTCycleID	The Cycle ID, if any, assigned to the job.
Job.ExecutionStart	The date/time at which the job started
Job.ExecutionFinish	The date/time at which the job ended, if it has

Job Step

Variables defined for job steps. These variables are defined only if a step is currently executing.

Variable	Description
StepStatus	The current status of the Step.
ADTJobStep	The OID of the step.
ADTStepNumber	The step number.
ADTStepDescription	The description of the step, including name if defined, and task description.
ADTStepNotes	The Description/Notes information for the step
ADTStepName	The name of the step, if defined.
ADTJobCheckpoint	The current checkpoint for the step.
ADTStepRestartCount	The restart attempt number. Will be 0 for first execution attempt, then incremented each time the job is restarted by a Restart action.
ADTServerPath	The path in which the adTempus server executable and related tools are located.
JobStep.ExecutionStart	The date/time at which the current step started
JobStep.ExecutionFinish	The date/time at which the current step ended, if it has

Task: E-Mail

Variables set by E-Mail Processing Tasks

Variable	Description
EmailProcessing.MessageLocation	Semicolon-delimited list of directories created by message save operations
EmailProcessing.SelectedMessageCount	The number of messages selected by the message filter

Task: File Operation

Variables set by File Transfer and File Compression Tasks

Variable	Description
FileOperationTask.ProcessedFileCount	The number of messages successfully processed by the task

Task: Process Termination

Variables set by Process Termination Tasks

Variable	Description
ProcessTermination.KilledProcessCount	The number of processes matching the

Variable	Description
	target process name that were successfully terminated
ProcessTermination.FailedToKillProcessCount	The number of processes matching the target process name that were not successfully terminated
ProcessTermination.KilledChildProcessCount	The number of processes terminated that were child processes of target processes
ProcessTermination.FailedToKillChildProcessCount	The number of processes not terminated that were child processes of target processes
ProcessTermination.KilledChildProcesses	Comma-separated list of the names of the child processes that were terminated
ProcessTermination.FailedToKillChildProcesses	Comma-separated list of the names of the child processes that could not be terminated

Task: Program Execution

Variables set by Program Execution Tasks

Variable	Description
ProcessExitCode	The exit code from the process run by the task.
ADTTaskResult	The exit code from the process run by the task.

Task: Script Execution

Variables set by Script Execution Tasks

Variable	Description
ScriptResult	The result returned by the script.
ADTTaskResult	The result returned by the script.

Task: Web Request

Variables set by Web Request Tasks.

Variable	Description
WebRequest.RequestURL	The URL requested by the task.
WebRequest.StatusCode	The HTTP status code returned by the remote server.
WebRequest.ResponseFileName	The name of the file where the server response (file or page) was saved, if the save to file option was selected.

Trigger: Computer Monitor

Variables set by Computer Monitor triggers

Variable	Description
ComputerMonitor.ErrorMessage	The error message if the connection has failed.
ComputerMonitor.ConnectionStatusCode	The numeric error code if the connection has failed, or 0 if it succeeded.
ComputerMonitor.Action	The action that caused the trigger: ConnectionFailed or ConnectionRestored.
ComputerMonitor.FailureCount	Total number of failed connection attempts since the last successful connection,
ComputerMonitor.Target	The target URL
ComputerMonitor.LastSuccessTime	The date/time of the last successful connection, in UTC

Trigger: File

Variables set by File Triggers

Variable	Description
FileTrigger.FileName	The name of the file(s) that matched the selection criteria. If the trigger is configured to list all matching files, and not configured to trigger separately for each file, the FileName variable will contain a list of all matching files, separated by semicolons.
FileTrigger.FileSize	The size of the file.
FileTrigger.FileCreationTime	The file's creation timestamp, in UTC.
FileTrigger.FileLastAccessTime	The file's last access timestamp, in UTC.
FileTrigger.FileLastWriteTime	The file's last write timestamp, in UTC.
FileTrigger.FileAttributes	The file's file attributes
FileTrigger.FileAction	The reason the file was selected: FileCreated, FileDeleted, FileExists, FileModified, or FileNoExist.

Trigger: Process

Variables set by Process Triggers

Variable	Description
ProcessTrigger.ProcessName	The name of the process
ProcessTrigger.ProcessID	The Windows process ID for the process
ProcessTrigger.ProcessEvent	The action that caused the trigger: "TriggerProcessStarted", "TriggerProcessEnded", "TriggerProcessMemExceeds"

Variable	Description
ProcessTrigger.ProcessEventDescription	Description of the action that caused the trigger: "started", "ended", or "exceeded memory threshold"

Trigger: Event Log

Variables set by Event Log Triggers

Variable	Description
EventLog	The name of the log where the event was reported.
EventLogMonitor.EventSource	The source of the event.
EventLogMonitor.EventCategory	The event category.
EventLogMonitor.EventCategoryDescription	The text description of the event category.
EventLogMonitor.EventType	The type/severity of the event.
EventLogMonitor.EventTypeName	The text description of the type/severity of the event.
EventLogMonitor.EventTimestamp	The timestamp of the event.
EventLogMonitor.EventID	The event ID.
EventLogMonitor.EventMessage	The message logged for the event.
EventLogMonitor.EventKeywords	Comma-delimited list of keywords for the event.

Trigger: E-Mail

Variables set by E-Mail Triggers

Variable	Description
EmailProcessing.SelectedMessageCount	The number of messages selected by the message filter
EmailProcessing.MessageId	The message ID of the selected e-mail message. Set only if the trigger is configured to trigger separately for each message.
EmailProcessing.MessageSubject	The subject of the selected e-mail message. Set only if the trigger is configured to trigger separately for each message.
EmailProcessing.MessageLocation	The name(s) of the files folders where the e-mail messages have been saved, if message saving is selected for the trigger. If multiple messages are selected, the EmailMessageLocation will contain multiple values, separated by semicolons. Each message is saved in a uniquely-named folder.

Alert Notification

Variables that are available in alert notification subjects and messages

Variable	Description
computername	The name of the computer issuing the alert
adTempus.InstanceName	The adTempus instance name.
adTempus.ServerAndInstanceName	The name of the server and instance, e.g., "computername\instancename". If this is the default instance, will be only "computername".
message	The alert message text
messageclass	The alert class (category)
messageid	The alert message ID
messagetype	The alert severity
messagecode	The message resource ID string
timestamp	The message timestamp
jobname	The name of the job associated with the alert

Related Concepts

Job Variables 88

Reference

Job Variable Properties 412

How To...

How To: Create a Basic Job

This example shows how to create a simple job to run the "Notepad" program from adTempus.

To create the job:

1. Select the Job Group that you want to create the job in. Right-click the group's name and select **New job** from the pop-up menu. (If you don't have any Groups defined, or want to create the job in the root group, right-click the **Jobs** node.)
2. The Job Properties window will open. Though adTempus offers many options, a simple job requires only a few things.
3. Enter a **Name**. Give the job a unique, descriptive name, such as "Run Notepad."
4. Enter the **User Account**. Type the user ID for the Windows user account the job should run under. If the account is a domain account, include the domain, for example `mydomain\myuserid`. When you leave the User Account box, you will be prompted to enter the password for this account. For this example, use your own user ID.
5. Click to the Steps page.
6. Click **Add...** and select the "Execute a program, batch file, etc." task type. This creates a Program Execution Task, which runs a program or batch file and is the most commonly-used kind of task.
7. The Program Execution Task properties window will open.
8. Next to the **Target** box, click the "**...**" button to browse for the file to execute. Browse to the Windows directory (usually "c:\windows") and select `notepad.exe`.
9. Click **OK** to save the task properties and return to the job properties.
10. Click **OK** to save the job.
11. The new job will now appear in the folder.

Because this job does not have any Triggers assigned to it, adTempus will run it only if you request it. To run the job:

1. Right-click the job's name and select the **Run** command. The Execute Job window will be shown.
2. Check the **Make the job visible on my desktop** option.
3. Click **OK** to start the job, then click **Close** to acknowledge the informational message.
4. The job's status in the Console will be updated to show that it is running, and the Notepad program will open on your desktop.

This example assumes you are running the adTempus Console on the same computer where the adTempus server is installed. If you are not, or if you are connected to that computer through a Remote Desktop or Terminal Services connection, you may not be able to see the Notepad program running on your desktop. In this case you will need to terminate the job by right-clicking it and selecting the **Terminate** command.

4. Close Notepad. The job's status will be updated to show that the job has finished.

Next Steps

Most jobs will have a Trigger so that adTempus will run them automatically. The most common trigger is the Schedule Trigger, which runs the job at specified days and times. We'll add a simple Schedule Trigger to the job you just created.

1. Double-click the job to display its properties, then click to the Triggers page.
2. Click the **Add...** button and select "Execute according to a schedule." The Execution Schedule Properties window will open.
3. On the **Date Selection** page, check the **Trigger every 1 days** option. These settings will cause the job to run every day.
4. Click to the **Time Selection** page.
5. Select the **Trigger at these times** option.
6. Below the list, type in a time when the job should run. For this example, enter a time that is two or three minutes in the future.
7. Click the **+** button to add the time to the list.
8. Click **OK** to save the schedule properties. The Schedule Trigger properties window now appears, where you could set additional trigger options.
9. Click **OK** to return to the job.
10. Click **OK** again to save the job and return to the Console.
11. After a moment you will see a **Next Start** time for your job.
12. When the scheduled time arrives, the job will start.

How To: Send Notification Messages for Failed Jobs

Though the adTempus Console highlights failed jobs in the Jobs list and in the Failed Jobs view, it is common to configure important jobs to notify relevant personnel by e-mail if the jobs fail.

Such notification is configured using Responses that run Notification Actions. To configure the failure notification:

1. Edit the job and go to the Responses page in the Job Properties window.
2. Click **Add** to add a new Response.
3. In the Response Properties window, go to the **Events** section and click **Add**.
4. In the Response Event Properties window, select the "Job Failed" event and click **OK.**
5. Back in the **Response Properties window**, go to the **Actions** section and click **Add**.
6. In the **Select Action** window, select "Send a notification message" and click **OK**.
7. In the Notification Action Properties window, add the recipients (individuals or groups) that you want to notify when this job fails.
   - You can leave the Subject and Message blank, and adTempus will use a default message that indicates that the job has failed. Or you can provide your own subject and message.
   - If the job produces log files or other output, you can have adTempus attach those files to the notification message, using the settings on the **Attachments** page.
8. Click **OK** to return to the Response, then **OK** again to return to the job, and **OK** again to save the job.

adTempus will now send a notification message whenever the job fails for any reason.

If you want to use the same failure notification settings for multiple jobs, you put them in the same Job Group and configure the Response for the group instead of doing it for each job in the group. Responses for a group are configured on the Responses page of the Job Group Properties.

Related Concepts

Notification Action 291

Related Topics

Notification Task 181
Notification Action 291
Notification Recipients 93
Messaging Setup 417
Notification Recipient 303
Notification Group 309

How To: Link Jobs Together

Often you will want to link two or more jobs together so that one job is run when another finishes. In adTempus this is done using either Responses that run Job Control Actions or with

Job Triggers. In many cases the two approaches are interchangeable, but there are some differences.

When you use Job Control Actions:

- The Action is configured on the "source" job: If you want Job 2 to run when Job 1 succeeds, you add a Job Control Action to Job 1 to have it run Job 2 (the "target" job).
- You have more control over the circumstances in which the target job is run. For example, you could run the target job only if a step in the source job returns a particular exit code.
- You have more control over how the target job is run: you can specify which step(s) should be run, decide whether Responses should be run, ignore Conditions for the job, and control which target servers the job runs on.
- The Action can be blocked from running if the target job is held.

When you use Job Triggers:

- The Trigger is configured on the "target" job: You add a Job Trigger to Job 2 telling it to run when Job 1 succeeds.
- You can configure the trigger to fire based on more than one job, when all or any of the source jobs has run.
- You have more control over which instances of the source job can trigger the target.

Linking with a Job Control Action

In this example we will run the "Target" job when the "Source" job succeeds.

1. Create and save both the Source job and the Target job .
2. Edit the Source job and go to the Responses page of the Job Properties.
3. Click **Add** to add a new Response.
4. In the Response Properties window, go to the **Events** section and click **Add**.
5. In the Response Event Properties window, select the "Job Succeeded" event and click **OK**.
6. Back in the **Response Properties window**, go to the **Actions** section and click **Add**.
7. In the **Select Action** window, select "Control a job or job step" and click **OK**.
8. In the Job Control Action Properties window,
    a. Select the "Run a job" action.
    b. Under **Applies To**, select **Another job on the same computer**, then click **Select...** and select the Target job.

   c. Configure any other options as desired.

   d. Click **OK** to save the Action.

9. Click **OK** to save the Response, and **OK** again to save the job.

The jobs are now linked together (no configuration is required on the Target job).

In this example we want the Target job to run when the Source job succeeds. The link can also be based on many other criteria. For example, you could run a different Target job when the Source job fails. This is determined by the Event that you selected in step 5 above. See the Job Responses topic for information on the other Events that are available.

You could also run the Response from a particular step instead by using the Responses page of the step properties. This allows you to choose from additional events, such as running the Target job based on the exit code from a program that you run.

Linking with a Job Trigger

In this example we will again run the "Target" job when the "Source" job succeeds.

1. Create and save both the Source job and the Target job .

2. Edit the Target job and go to the Triggers page of the Job Properties.

3. Click **Add** to add a new Trigger.

4. In the **Select Trigger Type** window, select "Execute in response to another job" and click **OK**.

5. In the Job Trigger Properties window, click **Add** on the **Job Trigger** page to add a new trigger rule.

6. In the Job Trigger Target Rule window,

   a. Under **Trigger based on job**, select **Another job on the same computer**, then click **Select...** and select the Source job.

   b. Leave all other settings at their default

   c. Click **OK** to save the rule.

7. Click **OK** to save the trigger, and **OK** again to save the job.

The jobs are now linked together (no configuration is required on the Source job).

Related Concepts

Job Control Action....284
Job Trigger....236

How To: Perform an action if a job has not run by a certain time

Responses allow you to respond to problems or other events that occur during job execution. Sometimes it can also be useful to perform an action if a job does not run when expected.

For example, suppose you have a job that is executed each day when a data file is uploaded to the server (using a File Trigger). You want to be notified if that file has not arrived by 4:00 PM. You can accomplish this using a second job (the we'll call it the "Notification Job") that has a condition on the original job (the "File Processing Job").

Configure the Notification Job as follows:

- On the Triggers page, add a Schedule Trigger to schedule the job to run at 4:00 each day.
- On the Conditions page, select the **Skip the job** option.
- Add a Job Condition configured as follows:
  - **Depend on job**: Select the File Processing Job.
  - **Rule**: Job has not run.
  - **Instance**: Any since previous execution of current job.
  - **Condition wait**: Do not wait for condition to be met.
- On the **Steps** page, add a Notification Task to send notification to the appropriate recipient (s).

When the Notification Job is triggered at 4:00 each day, adTempus will check to see if the File Processing Job has run since the last time the Notification Job ran. If it has, the condition will not be satisfied, and the notification message will not be sent. If the File Processing Job has not run, the condition will be satisfied, and the notification message will be sent.

Some additional configuration may be required depending on your workflow. For example, supposed that the Files Processing Job has not run at 4:00 on Tuesday, so the Notification Job runs. You resolve the problem that caused the file to not be transferred on time, and the file is now transferred at 4:30, causing the File Processing Job to run. Now supposed that on Wednesday, there is again a problem with the transfer and the file does not arrive on time. However, when the Notification Job runs at 4:00, it finds the instance of the File Processing Job that ran on Tuesday at 4:30 (after the Notification Job had run). This causes the condition for the Notification Job to fail, so notification is not sent.

There are several ways to avoid this problem:

- Use a Cycle ID for the group that contains the two jobs, and change the Instance rule for the condition to "Any in same cycle."
- Add a File Trigger to the Notification Job that targets the File Processing Job, so that the Notification Job will always run right after the File Processing Job. The job won't do anything when it is triggered this way (because its condition will not be met), but this will

"reset" the dependency chain so that the File Processing Job must run again before the next scheduled execution of the Notification Job.

How To: Mirror jobs to Remote Agents

Note: This example assumes you have read the Distributed Scheduling Overview and have completed Distributed Scheduling Installation.

Suppose you have a set of jobs that you want to run each night on each of three different servers, using Distributed Scheduling. You have installed adTempus on the Master and each of the three Agents, and have set up Remote Agent definitions for each of the Agents as described in the Distributed Scheduling Setup topic. Now how do you get the jobs to run on the Agents?

1. Run the adTempus Console and connect to the Master adTempus server.
2. Right-click the **Queues** node and select the **New Queue** command. The Job Queue Properties window opens.
3. On the **Queue** page, give the Queue a name, such as "Mirrored Jobs."
4. Click to the **Distributed Scheduling** page.
5. In the **Agent Mode** section, select the **Mirror** option.
6. Uncheck the **Run on Master** option, unless you also want the job to run on the Master computer.
7. In the **Run on Remote Agents** section, click **Add**.
8. Select the first of your three Agents and click **OK**.
9. Click **OK** again to accept the default options.
10. Repeat steps 7 through 9 for each of the other two Agents.
11. Click **OK** to save the Queue.
12. Locate each of the jobs you want to mirror and drag it to the Queue you have just created (or edit the job and select the new Queue in the job's properties).

The jobs are now mirrored to the selected Agents, and will run independently on each Agent. When you select a job in the Console's Job List, the **Agents** tab will show you the status of the job on each Agent.

If you connect the Console to one of the Agent computers, you will see that the job appears in the Job List on that computer.

Disable Execution for All Jobs

Sometimes it is convenient to pause all job execution in adTempus, for example while making widespread configuration changes or troubleshooting system issues.

Four methods can be used to accomplish this:

Hold the Root job group

Beginning with adTempus 4 you can place a Job Group on hold, and this hold will affect all jobs in the group and its sub-groups. Therefore you can hold the Root job group (the "Jobs" node in the Console) to hold all jobs. To do so: Right-click the **Jobs** node and select **Hold > Disable all execution**.

Note that any user with sufficient permission can re-enable execution by releasing the group.

Shut down adTempus

You can stop adTempus from running jobs by simply shutting down the adTempus service. However, if the adTempus service is not running you will not be able to view or modify jobs or other configuration settings.

Disable job execution using a startup option

To avoid the possibility of another user releasing the held Root group, you can disable job execution by stopping the adTempus service and restarting it with the norun option.

Note that if the adTempus service is subsequently restarted (e.g., if the computer is restarted), the option will not be applied, and execution will resume.

Disable job execution using a Registry option

To ensure that execution remains disabled even if the adTempus service is restarted again without the "norun" option, you can disable execution through a Registry setting. To do so:

1. Stop the adTempus service.
2. Run the Registry Editor and go to the following key: `HKEY_LOCAL_MACHINE\Software\Arcana Development\adTempus\default\options`. (Note: If you are working with an adTempus instance other than the default, replace "default" with the appropriate instance name.)
3. Create a new DWORD value named "`Engine:RunJobs`" and set it to "0".
4. Start the adTempus service.

You can now connect to adTempus through the Console to view or modify job settings.

Note: This method disables all job execution. Jobs will not be triggered automatically, and will not execute if they are submitted manually.

To re-enable job execution, delete the "Engine:RunJobs" value and restart the adTempus service.

How To: Set and Retrieve Variables in Scripts

The values for Job Variables can be read or set at job execution time by any script run as part of the job. This provides even more flexibility in customizing your job's behavior at runtime.

Job Variables are available through the JobVariables collection exposed to all scripts.

When you set variables from a script, the values you set only apply within that instance of the job. To make persistent updates to variables you must use the Job Variable Update Task or Job Variable Update Action.

Related Concepts

Job Variables 88

Reference

Job Variable Properties 412

How To: Set the Checkpoint Using the adTempus API

A Microsoft.NET application can query or set the current checkpoint for a job using the ApplicationIntegration class. Use the GetCheckpoint or SetCheckpoint method.

How To: Specify Notification Subjects, Messages, and Severities at Run-Time

When you use a Notification Action, File Capture Action, or Notification Task, you specify the subject and message text to be used in the notification messages sent by adTempus.

In many cases it is useful to be able to customize the content of these messages at run-time. For example, if you have a notification message that gets sent when a job starts because of a File Trigger, you might want the name of the file to be included in the message.

This can be accomplished using Job Variable tokens in the message subject and body fields.

The predefined %DEFAULTMESSAGE% token can be used in Notification Actions. It expands to the default message for the event that triggered the action; the message describes the event that occurred (such as "Job XXX Failed").

Notification Severity

You can also override the notification severity using a script (run by a Script Condition, Script Execution Task, etc.). Simply set the **MessageSeverity** variable to the severity you want to use (1 through 9). When you make this change, it applies only to the next notification action taken by the job (i.e., the next Notification Action, Notification Task, or File Capture Action executed by the job).

If you want to override the severity for all subsequent notifications in the entire job, set the **MessageSeverityAll** variable instead.

How To: Set Environment Variables with a Script

The environment variables passed to a scheduled program can be modified using the Job Variables page of the Program Execution task. However, this only allows you to specify static values (that is, you must set the values at the time you set up the job).

In some cases you may want to set environment variables dynamically (at the time the job runs). You can do this using scripts, either by running a Script Action or a Script Execution task. This is done using the global JobVariables collection available to all scripts run by adTempus. When you set an environment variable this way, the changes apply to all steps executed after you set the value.

You can also set values using the Job Variable Update Task or Job Variable Update Action.