independIT Integrative Technologies GmbH
Bergstraße 6
D–86529 Schrobenhausen

BICsuite!Web

Benutzerhandbuch
Release 2.6

Dieter Stubler Ronald Jeninga

July 29, 2014

Copyright © 2014 independIT GmbH

Rechtlicher Hinweis

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung der independIT GmbH in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Inhaltsverzeichnis

Inhaltsverzeichnis i

Tabellenverzeichnis ix

Abbildungsverzeichnis xvi

1 Allgemein 1
- 1.1 Überblick . . . 1
- 1.2 Login . . . 2
- 1.3 Main-Desktop . . . 2
- 1.4 Fensteraufbau Objektfenster . . . 5
  - 1.4.1 Titelleiste . . . 5
  - 1.4.2 Navigator . . . 7
    - 1.4.2.1 Standard-Buttons . . . 7
  - 1.4.3 Editor . . . 9
    - 1.4.3.1 Standard-Buttons . . . 10
- 1.5 Werteauswahl . . . 15
  - 1.5.1 Werteauswahl durch ”Drop Down” Liste . . . 15
  - 1.5.2 Werteauswahl durch Auswahlbutton (Chooser) . . . 16
- 1.6 Standard Listen Handling . . . 17
  - 1.6.1 Hinzufügen einer Zeile . . . 17
  - 1.6.2 Löschen einer Zeile . . . 18
  - 1.6.3 Ändern von Zeilenwerten . . . 19
- 1.7 Copy und Paste und Zwischenablage . . . 20
  - 1.7.1 Verschieben von Objekten in der Hierarchie . . . 21
  - 1.7.2 Verknüpfung von Objekten . . . 23
- 1.8 Grafik-Legende . . . 26
  - 1.8.1 Darstellung der BICsuite Objekte . . . 26
  - 1.8.2 Darstellung der Parent-Child-Beziehungen in Abläufe . . . 26
  - 1.8.3 Darstellung der Abhängigkeiten zwischen Scheduling Entities 27
- 1.9 Rechte . . . 28
  - 1.9.1 Kurzbeschreibung . . . 28
  - 1.9.2 Ausführliche Beschreibung . . . 28
  - 1.9.3 Systemprivilegien . . . 30
- 1.10 Comments . . . 31
- 1.11 Standardfelder . . . 32

2 Exit State Definitions **33**
2.1 Bild . . . 33
2.2 Konzept . . . 33
2.2.1 Kurzbeschreibung . . . 33
2.2.2 Ausführliche Beschreibung . . . 33
2.3 Editor . . . 33

3 Exit State Mappings **35**
3.1 Bild . . . 35
3.2 Konzept . . . 35
3.2.1 Kurzbeschreibung . . . 35
3.2.2 Ausführliche Beschreibung . . . 35
3.3 Editor . . . 36

4 Exit State Profiles **39**
4.1 Bild . . . 39
4.2 Konzept . . . 39
4.2.1 Kurzbeschreibung . . . 39
4.2.2 Ausführliche Beschreibung . . . 39
4.3 Editor . . . 40

5 Exit State Translations **43**
5.1 Bild . . . 43
5.2 Konzept . . . 43
5.2.1 Kurzbeschreibung . . . 43
5.2.2 Ausführliche Beschreibung . . . 43
5.3 Editor . . . 44

6 Resource State Definitions **45**
6.1 Bild . . . 45
6.2 Konzept . . . 45
6.2.1 Kurzbeschreibung . . . 45
6.2.2 Ausführliche Beschreibung . . . 45
6.3 Editor . . . 45

7 Resource State Profiles **47**
7.1 Bild . . . 47
7.2 Konzept . . . 47
7.2.1 Kurzbeschreibung . . . 47
7.2.2 Ausführliche Beschreibung . . . 47
7.3 Editor . . . 47
7.3.1 Beispiel . . . 48

8 Resource State Mappings **51**

8.1 Bild . . . 51
8.2 Konzept . . . 51
8.2.1 Kurzbeschreibung . . . 51
8.2.2 Ausführliche Beschreibung . . . 51
8.3 Editor . . . 52

9 Import/Export **55**
9.1 Konzept . . . 55
9.2 Import/Export aus dem Main Desktop . . . 55
9.3 Import/Export Objektbezogen . . . 56

10 Named Resources **59**
10.1 Bild . . . 59
10.2 Konzept . . . 59
10.2.1 Kurzbeschreibung . . . 59
10.2.2 Ausführliche Beschreibung . . . 59
10.3 Editor . . . 60
10.3.1 Tab Properties bei Kategorien . . . 60
10.3.2 Tab Content . . . 61
10.3.3 Tab Properties bei Named Resource Definitionen . . . 61
10.3.4 Tab Parameters . . . 63
10.3.4.1 Tab Parameter Details . . . 63
10.3.4.2 Standard Parameters . . . 64
10.3.5 Tab Resources . . . 64
10.3.6 Tab Job Definitions . . . 65
10.3.7 Tab Properties bei Pooled Resources . . . 66
10.4 Selektor Named Resource . . . 67

11 Environments **69**
11.1 Bild . . . 69
11.2 Konzept . . . 69
11.2.1 Kurzbeschreibung . . . 69
11.2.2 Ausführliche Beschreibung . . . 69
11.3 Navigator . . . 70
11.4 Editor . . . 70
11.4.1 Tab Properties . . . 70
11.4.2 Tab References . . . 71

12 Footprints **73**
12.1 Bild . . . 73
12.2 Konzept . . . 73
12.2.1 Kurzbeschreibung . . . 73
12.2.2 Ausführliche Beschreibung . . . 73
12.3 Editor . . . 74

12.3.1 Tab Properties . . . 74
12.3.2 Tab References . . . 75

13 Jobservers and Resources 77
13.1 Bild . . . 77
13.2 Konzept . . . 77
13.2.1 Kurzbeschreibung . . . 77
13.2.2 Ausführliche Beschreibung . . . 77
13.3 Navigator . . . 78
13.4 Editor . . . 78
13.4.1 Tab Properties . . . 78
13.4.2 Tab Resources . . . 81
13.4.2.1 Tab Resource Detail . . . 82
13.4.2.2 Tab Parameters . . . 85
13.4.2.3 Tab Allocations . . . 86
13.4.2.4 Tab Tracing . . . 90
13.4.3 Tab Parameters . . . 91
13.4.4 Tab Config . . . 92
13.4.4.1 Standard Konfigurationsparameter . . . 93
13.4.5 Tab Env.Map . . . 94
13.4.6 Tab Logfile Pattern . . . 95
13.5 Pooled Resources . . . 96
13.5.1 Tab Resource Detail . . . 97
13.5.2 Tab Pooled Resources . . . 97
13.5.3 Tab Distributions . . . 99
13.5.4 Tab Tracing . . . 100
13.6 Resource Links . . . 101
13.6.1 Tab Resource Detail . . . 102

14 Batches und Jobs 103
14.1 Bild . . . 103
14.2 Konzept . . . 103
14.2.1 Kurzbeschreibung . . . 103
14.2.2 Ausführliche Beschreibung . . . 104
14.2.3 Empfohlene Konvention für Batch Objekte . . . 105
14.3 Navigator . . . 106
14.3.1 Pinning . . . 106
14.3.2 Folder Bookmarks . . . 106
14.3.3 Folder Search . . . 108
14.4 Editor für Folder . . . 108
14.4.1 Tab Properties . . . 108
14.4.2 Tab Content . . . 110
14.4.3 Tab Parameters . . . 110

14.4.4 Tab Resources . . . 110
14.5 Editor für Job Definitions . . . 112
14.5.1 Tab Properties . . . 112
14.5.2 Tab Run . . . 118
14.5.3 Tab Restart . . . 122
14.5.4 Tab Children . . . 123
14.5.4.1 Tab Child Details . . . 124
14.5.5 Tab Parents . . . 129
14.5.6 Tab Dependencies . . . 130
14.5.6.1 Tab Dependency Details . . . 131
14.5.7 Tab Dependents . . . 134
14.5.8 Tab Required Resources . . . 135
14.5.8.1 Tab Resource Details . . . 137
14.5.9 Tab Defined Resources . . . 141
14.5.10 Tab Parameters . . . 142
14.5.10.1 Tab Parameter Details . . . 142
14.5.10.2 Predefined Standardparameter des Laufzeitsystems 146
14.5.11 Tab References . . . 150
14.5.12 Tab Trigger . . . 150
14.5.12.1 Tab Trigger Details . . . 152
14.5.13 Tab Triggered by . . . 157
14.6 Job Hierarchy Navigation . . . 158

15 BICsuite!Web Users 161
15.1 Bild . . . 161
15.2 Konzept . . . 161
15.2.1 Kurzbeschreibung . . . 161
15.2.2 Ausführliche Beschreibung . . . 161
15.3 Navigator . . . 162
15.4 Editor . . . 162
15.4.1 BICsuite!Web Connection . . . 162
15.4.2 BICsuite!Server Connections . . . 164

16 BICsuite Server Users 165
16.1 Bild . . . 165
16.2 Konzept . . . 165
16.2.1 Kurzbeschreibung . . . 165
16.2.2 Ausführliche Beschreibung . . . 165
16.3 Navigator . . . 166
16.4 Editor . . . 166

17 Groups 169
17.1 Bild . . . 169
17.2 Konzept . . . 169

17.2.1 Beschreibung 169
17.3 Navigator 169
17.4 Editor 169
17.4.1 Tab Properties 169
17.4.2 Tab Manage Privileges 170
17.4.3 Grants 171

18 Time Scheduling **173**
18.1 Bild 173
18.2 Konzept 173
18.2.1 Kurzbeschreibung 173
18.2.2 Ausführliche Beschreibung 174
18.3 Navigator 174
18.4 Editor 174
18.4.1 Teilbereich Sub Schedule Details 178

19 Submit Batches und Jobs **185**
19.1 Bild 185
19.2 Konzept 185
19.2.1 Kurzbeschreibung 185
19.2.2 Ausführliche Beschreibung 185
19.3 Navigator 186
19.4 Editor 186

20 Bookmarks **189**
20.1 Bild 189
20.2 Konzept 189
20.2.1 Kurzbeschreibung 189
20.2.2 Ausführliche Beschreibung 189
20.3 Navigation 191

21 Running Master Jobs **193**
21.1 Bild 193
21.2 Konzept 193
21.2.1 Kurzbeschreibung 193
21.2.2 Ausführliche Beschreibung 193
21.3 Master Navigator 193
21.3.1 Master Navigator Query Maske 200
21.4 Detail Navigation 206
21.4.1 Tree Darstellung 206
21.4.2 Suchergebnis 206
21.4.3 Detail Navigation Query Maske 206
21.5 Detailmaske für Jobs 208
21.5.1 Buttons 208

21.5.2 Tab Properties . . . 212
21.5.3 Tab Run . . . 214
21.5.4 Tab Timestamps . . . 217
21.5.5 Tab Dependencies . . . 219
21.5.6 Tab Dependents . . . 223
21.5.7 Tab Resource(Req) . . . 226
21.5.8 Tab Resource(Def) . . . 228
21.5.9 Tab Parameters . . . 229
21.5.10 Tab Audit . . . 230

22 Search Running Jobs 233
22.1 Bild . . . 233
22.2 Konzept . . . 233
22.2.1 Kurzbeschreibung . . . 233
22.2.2 Ausführliche Beschreibung . . . 233
22.3 Navigator . . . 233
22.4 Detail Navigator Query Maske . . . 234

23 Kalender 237
23.1 Bild . . . 237
23.2 Konzept . . . 237
23.2.1 Kurzbeschreibung . . . 237
23.3 Ausführliche Beschreibung . . . 237
23.3.1 Tab Query . . . 238
23.3.2 Tab List . . . 239
23.3.3 Tab Day . . . 239
23.3.4 Tab Week . . . 239
23.3.5 Tab Month . . . 240

24 System Information 241
24.1 Bild . . . 241
24.2 Konzept . . . 242
24.2.1 Kurzbeschreibung . . . 242
24.2.2 Tab Config . . . 242
24.2.3 Tab System . . . 251
24.2.4 Tab Worker . . . 253
24.2.5 Tab Sessions . . . 254
24.2.6 Tab SME/Q . . . 256

25 Object Monitoring 259
25.1 Bild . . . 259
25.2 Konzept . . . 259
25.2.1 Kurzbeschreibung . . . 259
25.2.2 Ausführliche Beschreibung . . . 259

25.3 Navigator 261
25.4 Editor 261
25.4.1 Tab Properties 261
25.4.2 Tab Configuration 263
25.4.3 Tab Trigger 264
25.4.4 Tab Trigger Details 264
25.4.5 Tab Object Instances 266
25.4.6 Tab Object Events 267

Tabellenverzeichnis

1.1 Beschreibung der Icons 4

13.1 Lockmode-Matrix 89
13.2 Beschreibung der Jobserver Konfigurationsparameters 94

24.1 ParameterHandling Optionen 246
24.2 Übersicht der Trace Level 250
24.3 Session Status 256

Abbildungsverzeichnis

1.1 Login Bildschirm . . . 1
1.2 BICsuiteLauncher . . . 2
1.3 Main Desktop . . . 3
1.4 Beispiel eines normalen Dialogs . . . 5
1.5 Busy Meldung . . . 6
1.6 About Fenster . . . 6
1.7 Navigator mit hierarchischer Darstellung . . . 7
1.8 Suchen im Navigator . . . 9
1.9 Suchergebnis . . . 9
1.10 Editor mit Tabs . . . 10
1.11 Show Folderpath . . . 12
1.12 Hide Folderpath . . . 12
1.13 Show Hierarchypath . . . 13
1.14 Hide Hierarchypath . . . 13
1.15 Selektion umkehren; Anfangszustand . . . 14
1.16 Selektion umkehren; Resultat . . . 14
1.17 Auswahl eines Resource State Profiles; Anfangszustand . . . 16
1.18 Auswahl eines Resource State Profiles; Auswahlliste . . . 16
1.19 Auswahl eines Resource State Profiles; Resultat . . . 17
1.20 Beispiel Standard Listen Handling . . . 17
1.21 Listen Handling; Zufügen einer Zeile . . . 18
1.22 Listen Handling; Auswahl eines Wertes . . . 18
1.23 Listen Handling; Resultat . . . 19
1.24 Listen Handling; Löschen einer Zeile . . . 19
1.25 Listen Handling; Editieren . . . 20
1.26 Objekte verschieben; Anfangszustand . . . 21
1.27 Objekte verschieben; Editor Content Tab . . . 22
1.28 Objekte verschieben; Selektion . . . 22
1.29 Objekte verschieben; Ausschneiden . . . 22
1.30 Objekte verschieben; Auswahl des Zielordners . . . 23
1.31 Objekte verschieben; Einfügen . . . 23
1.32 Objekte verschieben; Resultat . . . 24
1.33 Verknüpfen von Objekten; Anfangssituation . . . 24
1.34 Verknüpfen von Objekten; Selektion . . . 25
1.35 Verknüpfen von Objekten; Einfügen . . . 25
1.36 Legende graphische Abbildung BICsuite Objekte . . . 26

1.37 Legende graphische Darstellung Parent-Child-Beziehung 26
1.38 Legende graphische Darstellung Abhängigkeitsbeziehung 27
1.39 Beispiel der Rechtevergabe 29
1.40 Darstellung der kumulierten Rechten 30
1.41 Erfassen von Kommentare 32

2.1 Exit State Definitions 33

3.1 Exit State Mappings 35
3.2 Exit State Mapping Editor 36

4.1 Exit State Profiles 39
4.2 Exit State Profile Editor 40

5.1 Exit State Translations 43
5.2 Exit State Translation Editor 44

6.1 Resource State Definitions 45

7.1 Resource State Profiles 47
7.2 Resource State Profile Editor 48
7.3 Zustandsdiagramm einer Resource 49

8.1 Resource State Mappings 51
8.2 Resource State Mapping Beispiel 52
8.3 Resource State Mapping Editor 52

9.1 Import/Export aus dem Main Desktop 55
9.2 Objekbezogener Import/Export 57

10.1 Named Resources 59
10.2 Kategorien; Properties Tab 60
10.3 Kategorien; Content Tab 61
10.4 Named Resources; Properties Tab 61
10.5 Named Resources; Parameter Tab 63
10.6 Named Resources; Parameter Details 63
10.7 Named Resources; Instanzen 64
10.8 Named Resources; Job Definitions Tab 65
10.9 Named Resources; Pool Properties 67
10.10 Named Resources; Pool Instanzen 67
10.11 Named Resources; Selektor 67

11.1 Environments 69
11.2 Environments Navigator 70
11.3 Environment Properties 71
11.4 Environment References 72

12.1 Footprints . . . 73
12.2 Footprint Properties . . . 74
12.3 Footprint References . . . 75

13.1 Jobservers and Resources . . . 77
13.2 Jobservers und Scopes Navigator . . . 79
13.3 Scope Properties . . . 79
13.4 Jobserver Properties . . . 80
13.5 Jobserver und Scope Ressourcen . . . 82
13.6 Resource Details . . . 83
13.7 Resource Parameters . . . 85
13.8 Resource Allocations . . . 86
13.9 Resource Tracing . . . 90
13.10 Jobserver und Scope Parameters . . . 91
13.11 Jobserver und Scope Konfiguration . . . 93
13.12 Jobserver Environment Mapping . . . 95
13.13 Jobserver Logfile Name Patterns . . . 95
13.14 Pooled Resources . . . 96
13.15 Pool Details . . . 97
13.16 Pooled Resource; aktuelle Distribution . . . 98
13.17 Pooled Resource; Distributionen . . . 99
13.18 Pooled Resource; Tracing . . . 100
13.19 Resource Links kopieren und erstellen . . . 101
13.20 Resource Link Information . . . 102

14.1 Batches und Jobs . . . 103
14.2 Batches und Jobs Navigator . . . 106
14.3 Batches und Jobs; Ansicht ohne Pin . . . 107
14.4 Batches und Jobs; Ansicht mit Pin . . . 107
14.5 Folder Properties . . . 108
14.6 Folder Content . . . 110
14.7 Folder Parameters . . . 110
14.8 Folder Resources . . . 111
14.9 Job Definition Properties . . . 112
14.10 Batchbeispiele . . . 113
14.11 Beispiel für Milestone . . . 114
14.12 Job Run Tab . . . 118
14.13 Job Tab Restart . . . 122
14.14 Batches und Jobs; Children Tab . . . 123
14.15 Batches und Jobs; Child Details . . . 124
14.16 Statische und dynamische Children . . . 125
14.17 Beispiel für Merge Mode . . . 127
14.18 Beispiel Ignored Dependencies . . . 129
14.19 Batches und Jobs; Parents Tab . . . 129

14.20 Batches und Jobs; Dependencies Tab . . . 130
14.21 Beispiel für Dependency Modes . . . 131
14.22 Batches und Jobs; Dependency Details . . . 132
14.23 Beispiel für Unresolved Handling . . . 133
14.24 Batches und Jobs; Dependents Tab . . . 135
14.25 Jobs; Required Resources Tab . . . 135
14.26 Batches und Jobs; Resource Requirement Details . . . 137
14.27 Beispiel für Sticky Handling . . . 139
14.28 Beispiel für Lastverteilung mit Sticky Handling . . . 139
14.29 Batches und Jobs; Defined Resources . . . 141
14.30 Batches und Jobs; Parameters Tab . . . 142
14.31 Batches und Jobs; Parameter Details . . . 143
14.32 Beispiel für Expression . . . 144
14.33 Batches und Jobs; References Tab . . . 151
14.34 Batches und Jobs; Trigger Tab . . . 151
14.35 Batches und Jobs; Trigger Details . . . 152
14.36 Beispiel für Immediate Merge Trigger . . . 154
14.37 Beispiel für Immediate Local Trigger . . . 154
14.38 Beispiel für Before Final Trigger . . . 155
14.39 Beispiel für After Final Trigger . . . 155
14.40 Beispiel für Finish Child Trigger . . . 156
14.41 Jobs und Batches; Triggered By Tab . . . 158
14.42 Batches und Jobs; Hierarchie Ansicht . . . 158
14.43 Hierarchie Ansicht mit Dependencies . . . 159
14.44 Hierarchie Ansicht mit mehrfachen Dependencies . . . 160
14.45 Zyklische Abhängigkeiten . . . 160

15.1 Benutzerverwaltung für das Web Frontend . . . 161

16.1 BICsuite Server Benutzerverwaltung . . . 165

17.1 Gruppenverwaltung . . . 169
17.2 Gruppeneigenschaften . . . 170
17.3 Manage Privilegien einer Gruppe . . . 170
17.4 Objektprivilegien einer Gruppe . . . 171

18.1 Time Scheduling . . . 173
18.2 Time Scheduling Navigator . . . 175
18.3 Time Scheduling Editor . . . 176
18.4 Repeat Driver . . . 179
18.5 Time Of Day Driver . . . 179
18.6 Range of Day Filter . . . 180
18.7 Day of Week Filter . . . 180
18.8 Day of Month Filter . . . 181

18.9 Week of Month Filter . . . 181
18.10 ISO Week of Month Filter . . . 182
18.11 Week of Year Filter . . . 182
18.12 Month of Year Filter . . . 183
18.13 Calendar Driver . . . 183
18.14 Calendar Filter . . . 184

19.1 Submit Batches und Jobs . . . 185
19.2 Submit Navigation . . . 186
19.3 Submit Editor . . . 186

20.1 Bookmarks . . . 189

21.1 Running Master Jobs . . . 193
21.2 Running Master Jobs Navigator . . . 194
21.3 Statusdiagramm von Batches und Jobs . . . 200
21.4 Running Master Jobs Querymaske . . . 201
21.5 Running Master Jobs Detailfenster . . . 206
21.6 Detail Navigation Query Maske . . . 207
21.7 Detailmaske für Jobs . . . 208
21.8 Confirm Maske . . . 209
21.9 Job und Batch Properties . . . 212
21.10 Batch und Job Run Information . . . 215
21.11 Batch und Job Timestamps . . . 218
21.12 Batch und Job Abhängigkeiten . . . 219
21.13 Batch und Job Ignore Dependencies Confirm Maske . . . 219
21.14 Beispiel für Recursive Ignore . . . 220
21.15 Beispiel für Job Only Ignore . . . 221
21.16 Batch und Job Dependents Tab . . . 223
21.17 Batch und Job Ignore Dependencies . . . 224
21.18 Job Resource Requirements . . . 227
21.19 Job Ignore Resource Requirement . . . 227
21.20 Batch und Job Defined Resources . . . 228
21.21 Batch und Job Parameters . . . 229
21.22 Batch und Job Auditing . . . 230

22.1 Search Running Jobs . . . 233
22.2 Search Running Jobs Query Maske . . . 234

23.1 Kalender . . . 237
23.2 Kalender Query Tab . . . 238
23.3 Kalender Tab List . . . 239
23.4 Kalender Tab Day . . . 240
23.5 Kalender Tab Week . . . 240

23.6 Kalender Tab Month 240

24.1 System Information 241
24.2 System Konfiguration 242
24.3 System Runtime Umgebung 251
24.4 Worker Thread Information 253
24.5 Session Information 254

25.1 Object Monitoring 259
25.2 Object Monitoring; Navigator 261
25.3 Object Monitoring Neuanlage 261
25.4 Object Monitoring Eigenschaften 262
25.5 Object Monitoring Konfiguration 263
25.6 Object Monitoring Trigger 264
25.7 Object Monitoring Trigger Details 265
25.8 Object Monitoring Object Instances 267
25.9 Object Monitoring Object Events 268

1 Allgemein

1.1 Überblick

Im folgenden Kapitel wird eine Einführung in die BICsuite!web-Oberfläche gegeben. Die allgemeinen Aktionen und Verfahrensweisen, welche unabhängig vom gerade ausgewählten Dialog sind, werden hier erklärt und näher erläutert.

Die BICsuite!web-Oberfläche ist ein Browser-basiertes Dialogsystem. Es ist keine Installation von zusätzlicher Client-Software nötig.

Zum Starten der BICsuite!web-Oberfläche muss ein Browser gestartet werden (z. B. Microsoft Internet Explorer). Um auf die BICsuiteOberfläche zu gelangen geben Sie bitte die URL ein, welche Sie bekommen haben. Es empfiehlt sich diese URL in Ihrem Browser als Favorit abzuspeichern. So können Sie die BICsuite!web-Oberfläche durch einen Klick starten.

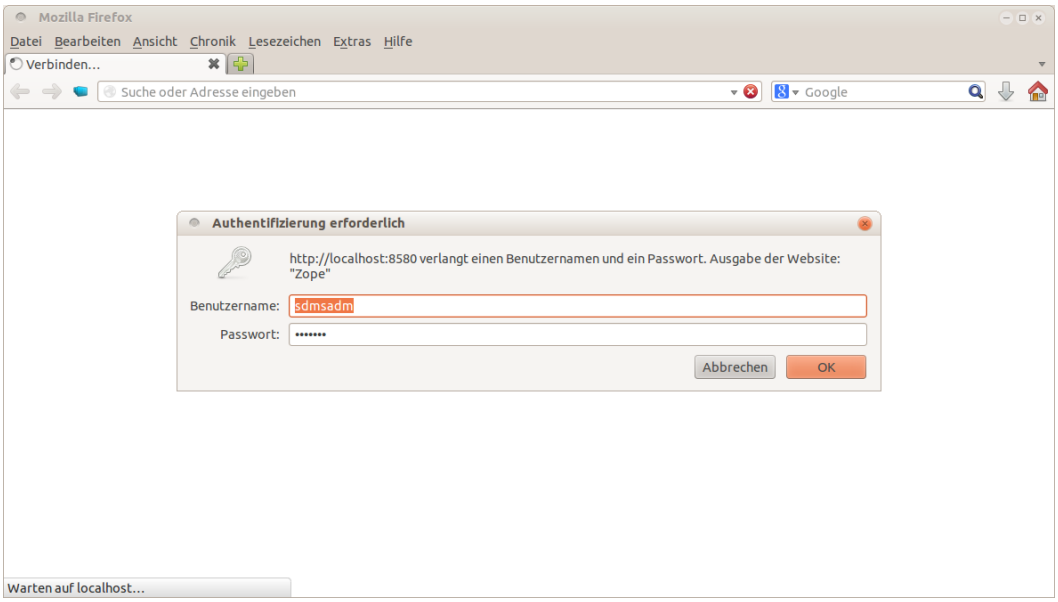


Abbildung 1.1: Login Bildschirm

1.2 Login

Nach der Eingabe der URL öffnet sich eine Login-Maske zur Passwortüberprüfung. Jeder Benutzer muss sich mit seiner Benutzerkennung und seinem Passwort identifizieren um mit BICsuite arbeiten zu können.
Die Eingabemaske für die Passwortabfrage ist in Bild 1.1 dargestellt, das genaue Aussehen kann je nach Browser etwas variieren.
Sie tragen hier Ihren persönlichen Benutzernamen und das dazugehörige Passwort ein und bestätigen Ihre Eingabe mit *OK*. Die Login-Daten bekommen Sie von Ihrem Systemadministrator.

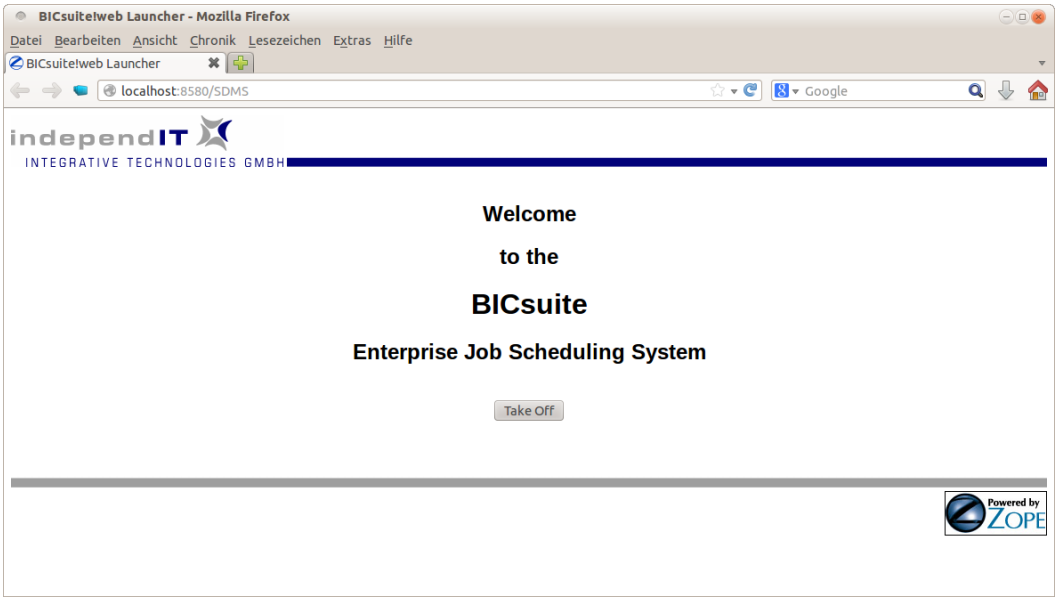


Abbildung 1.2: BICsuiteLauncher

Nach der korrekten Eingabe erscheint der BICsuiteLauncher. Starten Sie jetzt das Hauptmenü durch Drücken des *Take Off* Buttons.
Abhängig vom Browser wird der Launcher automatisch geschlossen. Sicherheitseinstellungen des Browsers können dies verhindern oder können zu einer Warnmeldung führen.
Nach dem Drücken des *Take Off* Buttons erscheint das Hauptmenü, der *Main-Desktop*.

1.3 Main-Desktop

Der *Main-Desktop* ist das Hauptfenster der BICsuite!web-Oberfläche. Mit diesem Fenster können Sie alle Funktionsdialoge der BICsuite direkt starten. Über das Optionsfeld "Connection" in der Kopfzeile kann gewählt werden, welche Server Verbindung für den nächsten zu öffnenden Funktionsdialog verwendet werden soll.

Main-Desktop

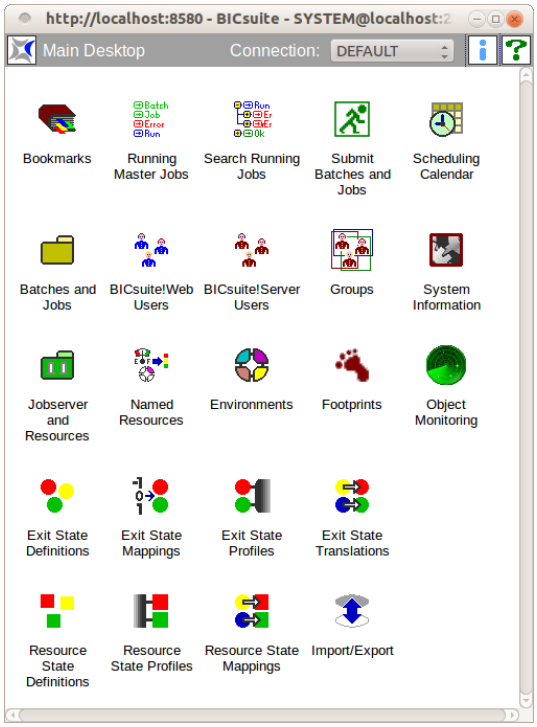


Abbildung 1.3: Main Desktop

Der *Main-Desktop* sieht aus wie in Bild 1.3.
Durch Anklicken der nachfolgend dargestellten Icons können die Dialoge zur Pflege oder Überwachung verschiedener Objekte aufgerufen werden:

Icon	**B**eschreibung
	Der *Bookmarks-Dialog* dient zum Anlegen, Ändern und Löschen von Sichten auf aktuell laufende Jobs.
	Der *Dialog "Running Master Jobs"* dient zur übersichtlichen Anzeige von aktuellen Master Jobs.
	Der *Search Running Jobs* dient zum Suchen nach laufenden Jobs nach bestimmten Kriterien.
	Der *Submit Jobs Dialog* dient zum Starten von Abläufe.
	Der *Dialog "Batches and Jobs"* dient zum Anlegen und Verwalten von Folders, Batches und Jobs und Milestones.
	Der *Dialog "BICsuite!Web Users"* dient zum Verwalten von BICsuite!web-Benutzern.

Wird fortgesetzt

Fortsetzung der vorherigen Seite

Icon	Beschreibung
	Der *Dialog "BICsuite!Server Users"* dient zum Verwalten von BICsuite!Server-Benutzern.
	Der *Groups-Dialog* dient zum Verwalten von BICsuiteBenutzergruppen.
	Der *Dialog "Jobserver and Resources"* dient zum Verwalten von Jobservern, Scopes und den zugehörigen Ressourcen
	Der *Dialog "Names Resources"* dient zum Anlegen, Ändern und Löschen von Named Resources.
	Der *Environments-Dialog* dient zur Definition von Laufzeitumgebungen für Jobs.
	Der *Footprint-Dialog* dient zum Verwalten von Footprints (Resource-Profile) im System.
	Der *Dialog "Exit State Definitions"* dient zur Definition und zur Verwaltung von Exit State-Namen.
	Der *Dialog "Exit State Mappings"* dient zur Verwaltung von Übersetzungen zwischen Rückgabewerten der Systemprozesse und logischen Exit State.
	Der *Dialog "Exit State Profile"* dient zum Verwalten von Profilen, also einer Zusammenfassung und Priorisierung von Exit States.
	Der *Dialog "Exit State Translation"* dient zum Verwalten der Übersetzungen von Exit States nach Exit States.
	Der *Dialog "Resource State Definition"* dient zur Definition und zur Verwaltung von Resource-Status-Namen.
	Der *Dialog "Resource State Profile"* dient zum Verwalten von Profilen, also Zusammenfassungen von Resource-Status.
	Der *Dialog "Resource State Mapping"* dient zum Verwalten der Übersetzungen von Exit State nach Resource-Statusänderung.
	Der *Dialog "Import/Export"* dient zum Importieren bzw. Exportieren von Dateien.
	Der *Dialog "Calendar"* bietet eine Übersicht der anstehenden Submits.
	Der *Dialog "SysInfo"* bietet Information über die Konfiguration sowie der aktuelle Zustand des Systems.
	Der *Dialog "Object Monitoring"* dient zum Verwalten und Überwachen von Objekten.

Tabelle 1.1: Beschreibung der einzelnen Icons des Hauptmenüs

1.4 Fensteraufbau Objektfenster

Nach dem Aufrufen eines Dialoges im *Main-Desktop* wird der Dialog in einem neuen Fenster angezeigt. Jedes Dialogfenster ist unabhängig von anderen Dialogfenstern und es können beliebig viele Dialoge zur selben Zeit offen sein.
Ein normaler Dialog sieht folgendermaßen aus:

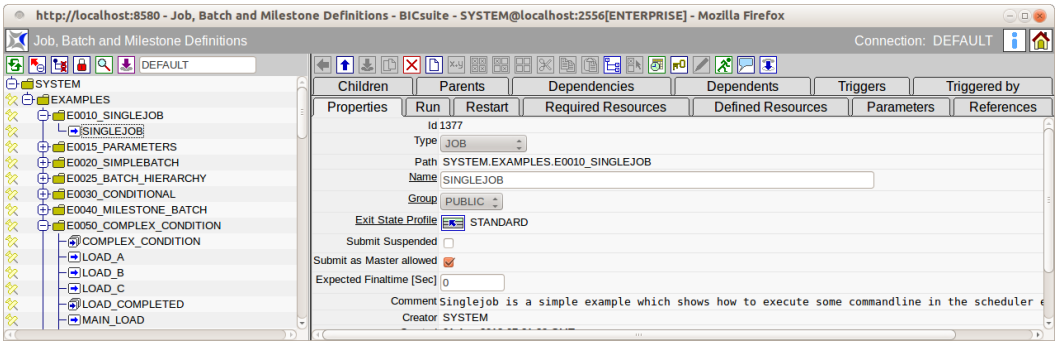


Abbildung 1.4: Beispiel eines normalen Dialogs

1.4.1 Titelleiste

Jeder Dialog besitzt im oberen Teil eine Titelleiste. Die Titelleiste enthält die nachfolgend beschriebenen Elemente.

independIT Icon/Sanduhr 
In der linken oberen Fensterecke, befindet sich das independIT Icon. Klickt man das independIT-Icon an, wird ein neues Fenster mit der Homepage der independIT GmbH angezeigt. Führt der Benutzer eine Aktion aus, müssen die neuen oder geänderten Daten an den BICsuite Server gesendet bzw. Daten vom Server geladen werden. Diese Aktivität wird durch das Sanduhr-Icon angezeigt. Solange das Sanduhr-Icon aktiv ist, ist keine weitere Aktion in dem Fenster möglich. Wird dennoch versucht eine Aktion durchzuführen, erscheint eine Fehlermeldung wie gezeigt in Bild 1.5.
In diesem Falle muss die Box mit *OK* bestätigt und auf das Ende der Datenübertragung vom Server gewartet werden. Verschwindet die Sanduhr und erscheint das independIT-Icon wieder, kann mit der Dateneingabe fortgefahren werden.

Name des Dialoges Nach dem independIT-Icon erscheint der aktuelle Name des Dialoges.

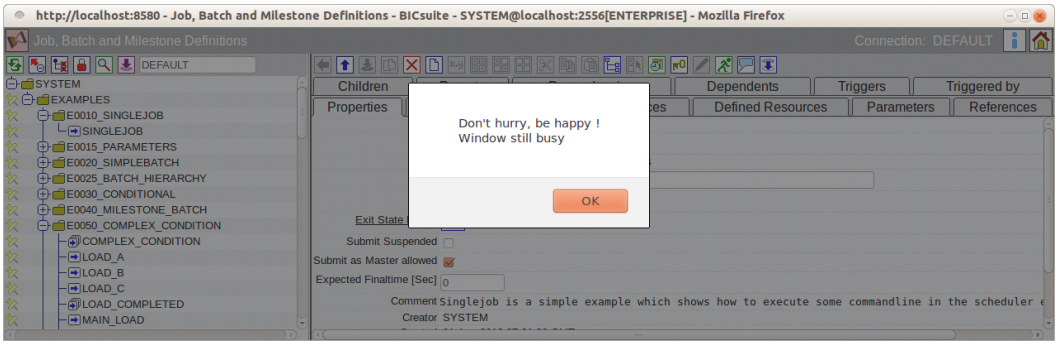


Abbildung 1.5: Busy Meldung

About Icon 

Mit diesem Button können Versions- und Systeminformationen angezeigt werden:

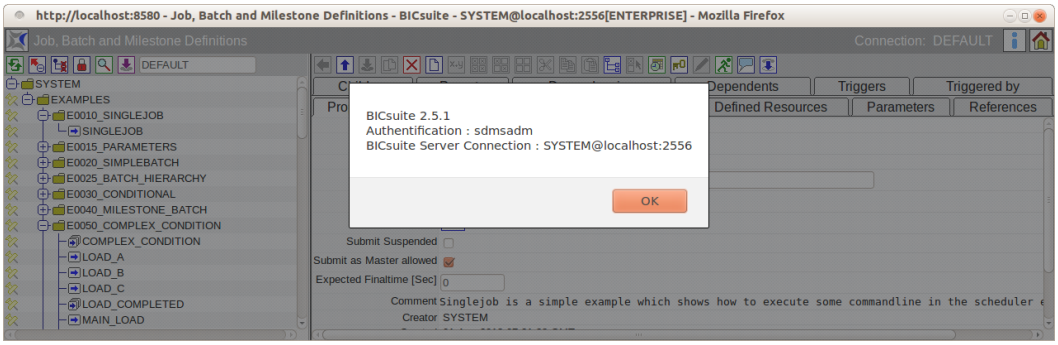


Abbildung 1.6: About Fenster

Folgende Informationen sind auf dem Fenster zu sehen.
BICsuite!web n.n.n: Die aktuelle Version der BICsuite!web-Oberfläche.
Authentication: Der Name des aktuell eingeloggten Benutzers.
BICsuite Server Connection: Die Verbindungsdaten für den BICsuite Server.

Help Icon 

Das Help Icon ist nur auf dem Main Desktop zu sehen und steht anstelle von dem Home Icon. Klickt man das Help Icon an, kommt man direkt in die Online-Dokumentation.

Home Icon 

Das Home Icon dient zum Aufrufen des Main Desktops. Ist das Fenster des Main Desktops minimiert oder im Hintergrund, wird es wiederhergestellt und nach vorne gehoben. Wurde es geschlossen wird ein neues Main Desktop-Fenster geöffnet.

1.4.2 Navigator

Der linke Teil eines Dialogfensters ist der Navigator. Er zeigt die bereits definierten Objekte an. Dies erfolgt entweder hierarchisch (falls die Art von Objekten hierarchisch anordbar sind z. B. Named Resources) oder als einfache Liste, wenn keine hierarchische Ordnung besteht. Die hierarchische Darstellung erfolgt in einer Baumstruktur und sieht dann aus wie in Bild 1.7.

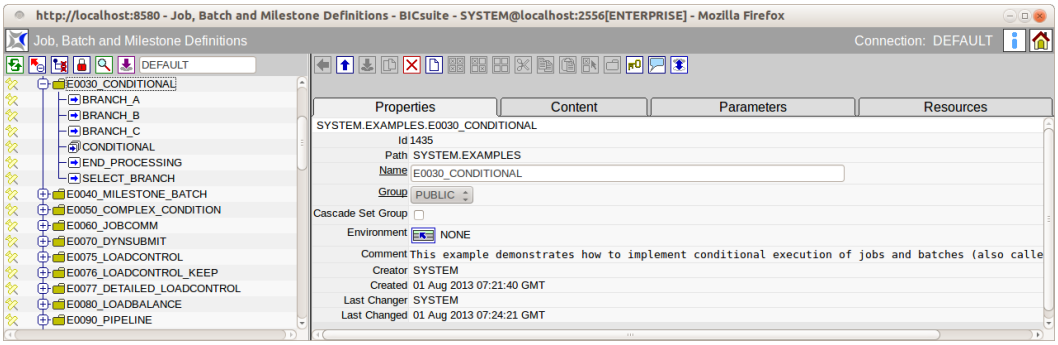


Abbildung 1.7: Navigator mit hierarchischer Darstellung

Handelt es sich bei einem in der hierarchischen Ansicht angezeigten Objekt um einen Container der noch andere Objekte enthalten kann, so wird dieser mittels eines Ordner-Icons, z. B.



dargestellt. Je nach Farbe und Darstellung handelt es sich um unterschiedliche Objekte (Folder, Scope, Named Resource etc.). Enthält ein Container darüber hinaus noch Unterobjekte so wird ein ⊕ oder ⊖ angezeigt. Mittels Anklicken dieser Icons, kann der Inhalt des Containers angezeigt ⊕ oder versteckt ⊖ werden.

1.4.2.1 Standard-Buttons

Über die nachfolgend beschriebenen Buttons, die sich über dem Navigator befinden, kann die Anzeige und das Anzeigen bzw. Verstecken von Inhalten gesteuert werden:

 Refresh Button

Der *Refresh* Button dient zum Auffrischen der Liste der Objekte in der Navigation. Nach dem Drücken dieses Buttons werden alle verfügbaren Objekte neu vom Server gelesen.

 Collapse All

Hiermit wird eine aufgeklappte Hierarchie, also eine Hierarchie in der Unterordner angezeigt werden (und der übergeordnete Ordner ein ⊟ besitzt), komplett wieder geschlossen. Das heißt, es sind nur noch die Objekte auf der obersten Hierarchiestufe sichtbar.

 Expand All

Hiermit kann eine geschlossene oder teilweise geöffnete Hierarchie komplett geöffnet werden. Das bedeutet, alle Container der obersten Ebene werden auf Untercontainer oder Unterobjekte untersucht und diese werden angezeigt. Dem Ordnersymbol wird das ⊟ Zeichen vorangestellt. Sollten die Untercontainer weitere Unterobjekte besitzen, so werden auch diese angezeigt, bis alle Objekte in der kompletten Hierarchie auf dem Bildschirm dargestellt werden. Handelt es sich um eine große Hierarchie mit vielen Objekten und Unterobjekten, kann dies eine verhältnismäßig lange Laufzeit zur Folge haben. Aus diesem Grunde ist der Button nicht bei jedem hierarchischen Objekttyp verfügbar.

 Show Leaves,  Hide Leaves

Diese beiden Buttons ermöglichen das Ein- und Ausblenden von Nicht-Containerobjekten in Hierarchien im Navigator. Es ist abwechselnd immer nur ein Button sichtbar.

 Show Locked,  Hide Locked

Diese beiden Buttons ermöglichen das Ein- und Ausblenden der Namen von Objekten auf die keine Leserechte vorliegen. Es ist abwechselnd immer nur einer der beiden Buttons sichtbar.

 Save View

Mit dem *Save* Button wird der momentane Zustand der Hierarchie im Navigator gespeichert.

 Suchen (Search)

Dieser Button dient zum Aktivieren der Suchmaske in der Navigation. Damit lässt sich in einer komplexen Hierarchie ein namentlich bekanntes Objekt sehr leicht finden. Nach Betätigen des *Suchen*-Buttons erscheint die Suchmaske:

 Start Find

Hiermit wird die Suche gestartet. Im abgebildeten Beispiel wird in der gesamten Hierarchie nach Named Resources gesucht, welche den Suchbegriff enthalten.
Treffer werden als Liste im Navigator angezeigt. Hierarchien werden nicht berücksichtigt. Das Ergebnis sieht z. B. folgendermaßen aus:
Hier wurde nach dem Begriff "Host" gesucht und drei Named Resources gefunden, welche den Begriff enthalten. Die Suche ist unabhängig von Groß- und Kleinschreibung des jeweiligen Suchbegriffes.

 Exit Find

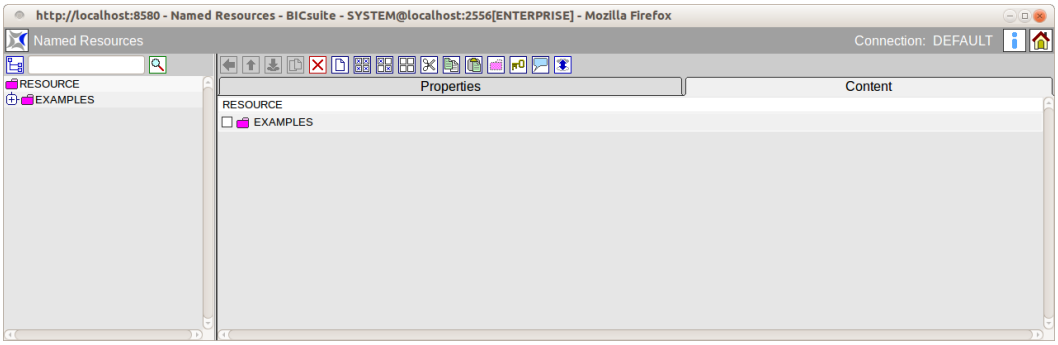


Abbildung 1.8: Suchen im Navigator

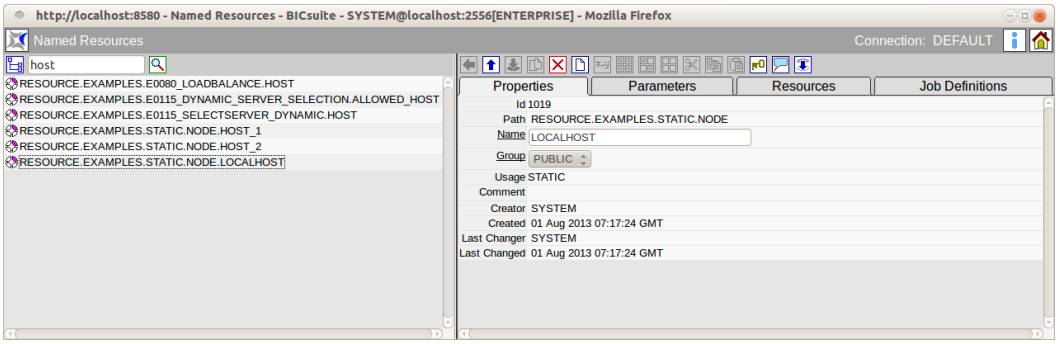


Abbildung 1.9: Suchergebnis

Hiermit wird die Suchmaske wieder verlassen. Nach Betätigung dieses Buttons wird wieder die normale Buttonleiste über dem Navigator angezeigt.

1.4.3 Editor

Der Editor ist der Dialog der zum Ändern oder Erstellen von Objekten dient. Wurde ein Objekt im Navigator ausgewählt, so erscheinen seine Daten im Editor und können hier geändert werden. Ein selektiertes Objekt kann über die Buttons am oberen Rand des Editors gespeichert, gelöscht oder dupliziert werden. Mit dem *Neu*-Button kann ein neues Objekt angelegt werden.
Die Daten innerhalb des Editors werden entweder komplett innerhalb eines Fensters angezeigt oder sind über unterschiedliche Tabs logisch gruppiert.
Die Anzahl der Tab Sheets in einem Dialog ist abhängig von der Art des gerade in der Navigation gewählten Objektes. Es kann innerhalb einer Aktion (Änderung, Neuanlage eines Objektes) beliebig oft und in beliebiger Reihenfolge zwischen den einzelnen Tabs gewechselt werden. Eine Zwischenspeicherung ist nicht nötig, da alle Daten (auch die Daten der nicht sichtbaren Tabs) im Speicher vorgehalten werden. Erst beim Betätigen des *Speichern*-Buttons werden die Daten aller Tabs zum

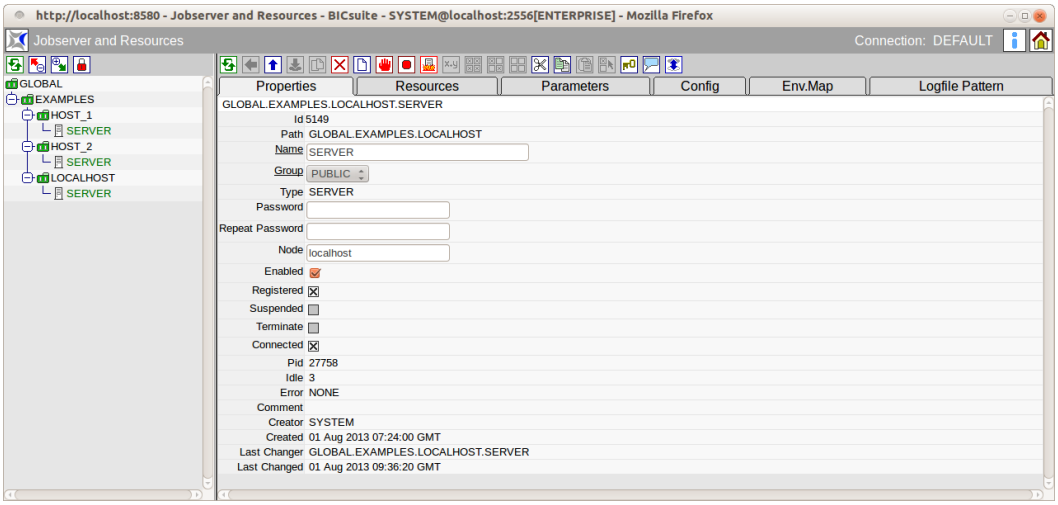


Abbildung 1.10: Editor mit Tabs

BICsuite Server übertragen und gespeichert. Falls Änderungen durch den *Cancel* Button rückgängig gemacht wurden, werden alle geänderten Daten verworfen, auch die auf den nicht sichtbaren Tabs.
Die Auswahl eines Tab Sheets erfolgt einfach über das Anklicken des jeweiligen Reiters, auf dem der Name des Tab Sheets steht. Die Daten des gewählten Tabs erscheinen anschließend auf der Maske und der gewählte Tab wird graphisch in den Vordergrund gehoben und hellblau markiert. Die nicht aktiven Tabs erscheinen dunkelblau.
Die "Editor" Buttonleiste zeigt je nach Tab unterschiedliche aktive und inaktive Buttons.

1.4.3.1 Standard-Buttons

Folgende Standard-Buttons sind bei vielen Dialogen vorhanden und sind in ihrer Handhabung in jedem Dialog identisch oder vergleichbar.

 Cancel

Dieser Button dient zum Widerrufen von Aktionen. Hiermit können alle noch nicht gesicherten Änderungen rückgängig gemacht werden. Das bedeutet alle Änderungen, welche seit dem Laden oder dem letzten Speichern durchgeführt worden sind, werden verworfen.
Der *Widerrufen*-Button ist nur aktiv falls eine Änderung durchgeführt wurde. Änderungen werden erst erkannt sobald das erste Feld, in dem eine Änderung durchgeführt wurde, verlassen wird. Ist nur ein Feld auf der Maske vorhanden, so muss die Tabulatortaste gedrückt werden um die Änderung zu akzeptieren. Anschließend schaltet sich der *Widerrufen*-Button aktiv.

Um ein Verlust der Eingabe auszuschließen, folgt nach Betätigung des Buttons noch eine Sicherheitsabfrage. Sollen die Änderungen wirklich verworfen werden, muss die Abfrage mit OK bestätigt werden, ansonsten kann mit Drücken von *Abbrechen* zum vorherigen Dialog zurückgekehrt werden und die gemachten Änderungen bleiben erhalten.

 Up

Mit dem *Up* springt man eine Hierarchieebene nach oben.

 Save

Dieser Button dient zum Speichern einer durchgeführten Änderung. Der *Speichern*-Button wird nur aktiv, wenn tatsächlich Daten geändert wurden. Änderungen werden erst erkannt, wenn das geänderte Eingabefeld verlassen wird.

 Clone

Dieser Button dient zum Erstellen eines neuen Objektes, welcher alle Eigenschaften des aktuell gewählten besitzt. Um den Button zu aktivieren, muss ein neuer Name für das Objekt eingetragen werden. Durch Betätigen des Buttons wird ein neues Objekt mit diesem Namen angelegt und im Navigationsfenster angezeigt.

Achtung: Falls statt des *Duplizieren*-Buttons der *Speichern*-Button betätigt wird, wird nur der Name des Objektes geändert, das heißt es wird kein neues Objekt erzeugt.

 Drop

Dieser Button dient zum Löschen des gesamten Objektes. Vor dem Löschen erfolgt eine Sicherheitsabfrage.

Wenn Sie das Objekt wirklich löschen wollen, bestätigen Sie dies mit dem *OK*-Button. Falls Sie den Vorgang abbrechen wollen, drücken Sie den *Abbrechen*-Button.

Nach der Bestätigung wird das Objekt vom Server gelöscht. Es gibt nun keine Möglichkeit mehr das Löschen rückgängig zu machen. Falls es sich um ein versehentliches Löschen handelt, muss das gelöschte Objekt neu angelegt werden.

Innerhalb von Listen können Selektionen ebenfalls gelöscht werden. Das Vorgehen ist hierbei folgendermaßen. Zuerst müssen die zu löschenden Zeilen einzeln oder über die *Selektions*-Buttons selektiert werden. Anschließend wird der *Drop*-Button betätigt. Nach der Bestätigung der Sicherheitsabfrage werden alle selektierten Einträge gelöscht.

 New

Dieser Button dient zum Anlegen eines neuen Objektes. Können innerhalb eines Dialoges unterschiedliche Objekte angelegt werden, so muss zuerst der Typ des neuen Objektes ausgewählt werden. Anschließend erscheinen die Felder und Tabs des entsprechenden Objekttyps. Ist keine Auswahl notwendig, erscheint der Editor sofort nach Betätigung des Buttons. Wenn ein Editorfenster noch ungespeicherte Änderungen enthält, ist der Button deaktiviert, damit ein versehentliches Löschen der geänderten Werte vermieden wird.

 Show Folderpath,  Hide Folderpath

Bei diesen Buttons handelt es sich um Schalter. Wurde der Button *Show Folderpath* gedrückt, so werden Scopes, Submitted Entities, Scheduling Entities und Folder mit der kompletten übergeordneten Hierarchie angezeigt. Die Ebenen werden jeweils durch ein Punkt getrennt. Nach Drücken des Buttons sieht die Anzeige folgendermaßen aus:

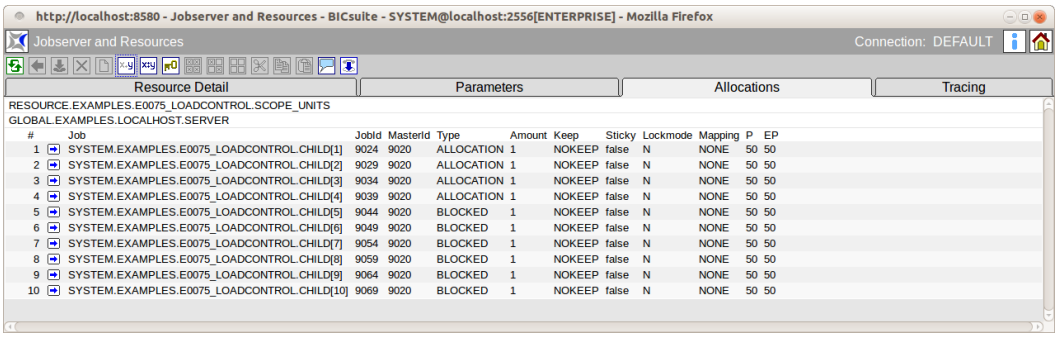


Abbildung 1.11: Show Folderpath

Wird der Button *Hide Folderpath* gedrückt, werden die bisher sichtbaren übergeordneten Ordner nicht mehr angezeigt. Die Anzeige folgendermaßen aus:

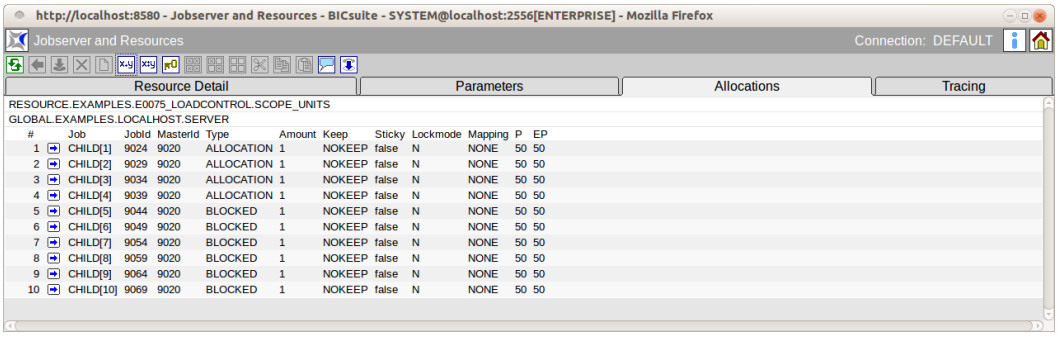


Abbildung 1.12: Hide Folderpath

 Show Hierarchypath,  Hide Hierarchypath

Bei diesen Buttons handelt es sich um Schalter. Wurde der Button *Show Hierarchypath* gedrückt, so wird das Ablaufobjekt mit der kompletten übergeordneten Parent-Child-Hierarchie angezeigt. Hat ein Ablaufobjekt einen Parent, wird dieser im Namen mit angezeigt und so weiter bis die oberste Ebene erreicht ist. Die Ebenen werden jeweils durch Doppelpunkt getrennt.
Wird der Button *Hide Hierarchypath* gedrückt, so wird die bisher sichtbare Hierarchie nicht mehr angezeigt. Nach Drücken des Buttons sieht die Anzeige folgender-

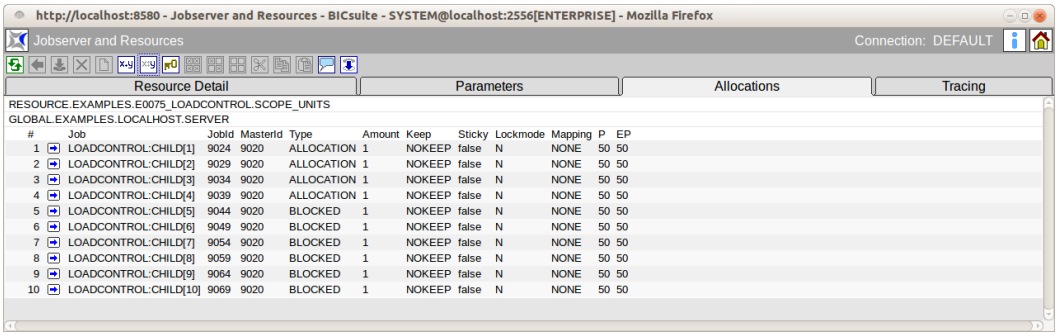


Abbildung 1.13: Show Hierarchypath

maßen aus:

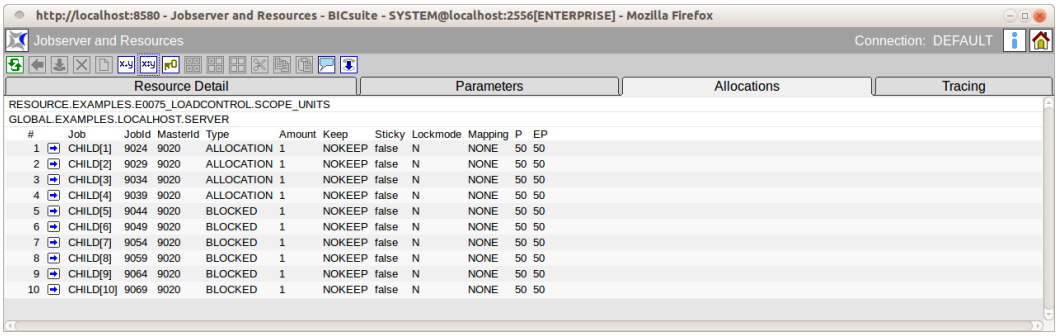


Abbildung 1.14: Hide Hierarchypath

 Select All

Dieser Button dient zum Selektieren aller Objekte. Das heißt, es werden alle Objekte, die aktuell im Navigator oder im ”Editor” Fenster angezeigt werden (je nachdem in welcher Buttonleiste sich der Button befindet), ausgewählt. Anschließend sind alle Objekte mit einem Kreuz ☒ vor Ihrem Namen markiert.
Nachdem alle Objekte selektiert wurden, können diese nun ausgeschnitten, kopiert oder gelöscht werden.

 Selektion umkehren (Toggle Selection)

Dieser Button dient zum Umkehren der aktuellen Selektion. Das heißt, alle selektierten Objekte werden deselektiert und alle noch nicht selektierten Objekte werden selektiert. Anschließend können nun die selektierten Objekte gelöscht, ausgeschnitten oder kopiert werden.
Das Verhalten wird in den folgenden beiden Screenshots verdeutlicht.
Nach dem Betätigen des Toggle Selection Buttons ist die Selektion ”umgekehrt”.

 Deselect All

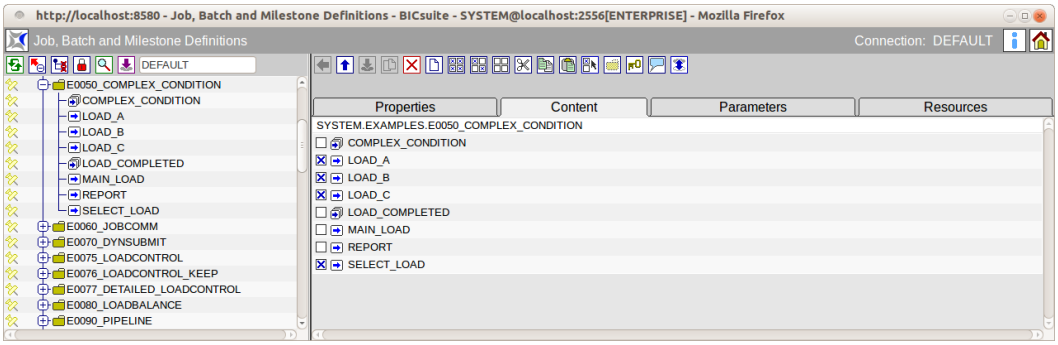


Abbildung 1.15: Selektion umkehren; Anfangszustand

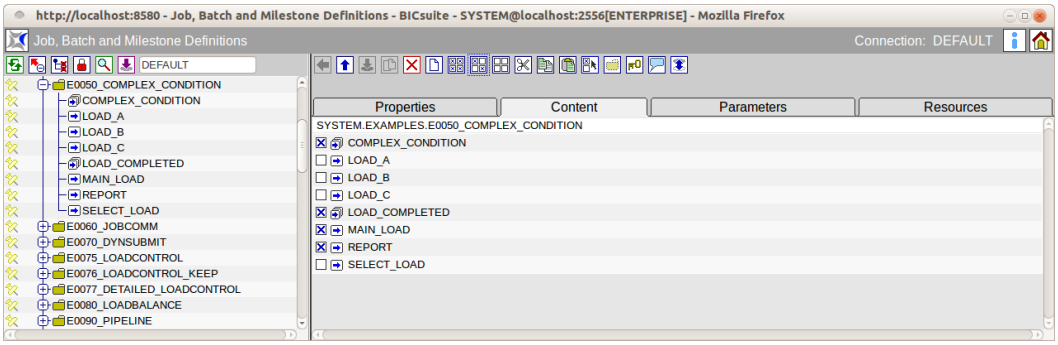


Abbildung 1.16: Selektion umkehren; Resultat

Dieser Button dient zum Aufheben der Selektion. Die Markierungen an den selektierten Einträgen werden entfernt. Hiermit kann eine bestehende Selektion wieder rückgängig gemacht werden.

 Cut

Dieser Button dient zum Ausschneiden von Objekten aus einer Liste oder einem Baum. Die Objekte werden in die BICsuite Zwischenablage kopiert. Die Objekte sind nur zum Ausschneiden vorgemerkt, sie verschwinden allerdings aus der aktuellen Ansicht. Durch Betätigen des *Einfügen*-Buttons können die Objekte an anderer Stelle wieder eingefügt werden.

 Copy

Mit dem *Kopieren*-Button können Objekte in die BICsuite Zwischenablage kopiert werden, um sie an anderer Stelle mittels des *Einfügen*-Buttons wieder einzufügen.

 Paste

Dieser Button dient zum Einfügen von vorher selektierten und geeigneten Objekten in einer Liste. Die Objekte müssen vorher innerhalb eines anderen Dialoges

oder Pop Up-Fensters selektiert werden.

 Select

Dieser Button dient zum Öffnen eines Fensters, in welchem für eine Liste geeignete Objekte selektiert und mit dem *Kopieren*-Button in die Zwischenablage kopiert werden können. Die Objekte in der Zwischenablage können dann mit dem *Einfügen*-Button in die Liste eingefügt werden.

 Time Scheduling

Dieser Button dient zum Aufrufen des Time Scheduling-Dialoges. Er wird nur angezeigt, falls ein Master Submittable Objekt im Navigationsbildschirm gewählt wurde.

 Grants

Mit diesem Button wird der Dialog zur Verwaltung der Zugriffsrechte geöffnet.

 Edit

Der *Edit* Button schaltet ein Editiermodus ein. Was editiert wird, ist abhängig vom jeweiligen Kontext.

 Import/Export

Mit diesem Button können Dateien importiert und exportiert werden.

 Export

Mit dem *Export* Button werden Dateien exportiert.

1.5 Werteauswahl

1.5.1 Werteauswahl durch ”Drop Down” Liste

Bei manchen Eingabefeldern ist die Auswahl aus einer Menge von vorgegebenen Werten erforderlich. In diesen Fällen wird ein *Drop Down* Feld angezeigt, dessen Liste alle möglichen Werte beinhaltet.
Ein *Drop Down* Feld sieht folgendermaßen aus:



Wird der Button neben dem Feld angeklickt oder mittels Tastaturbefehl ALT-Pfeil-Unten die ”Drop Down” Liste ausgeklappt, ist eine Einfachauswahl in der Liste möglich.
Die ausgeklappte ”Drop Down” Liste kann folgendermaßen aussehen:



Durch Anklicken des gewünschten Wertes wird dieser in das Feld übernommen. Alternativ ist es auch möglich, die Werte mittels der Pfeil-Unten- bzw. Pfeil-Oben-Tasten auszuwählen.

1.5.2 Werteauswahl durch Auswahlbutton (Chooser)

Wird die Liste der verfügbaren Werte erst zur Laufzeit vom Server gelesen, kommt der *Auswahl*-Button zum Einsatz. In der nachfolgenden Abbildung kann ein Resource State Profile mittels des *Auswahl*-Buttons gewählt werden:

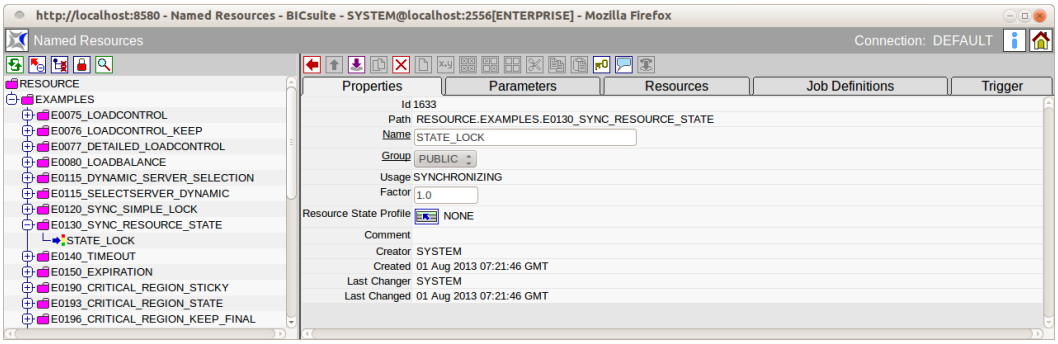


Abbildung 1.17: Auswahl eines Resource State Profiles; Anfangszustand

Nach dem Betätigen des *Auswahl*-Buttons



erscheint im Editor eine Liste von möglichen Objekten.

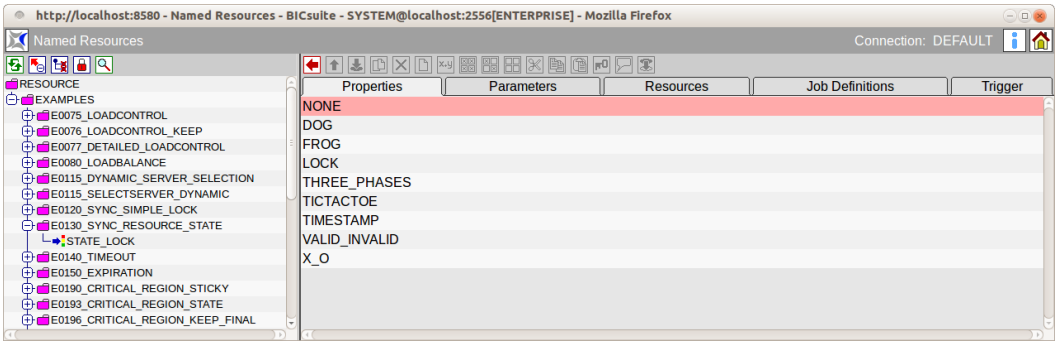


Abbildung 1.18: Auswahl eines Resource State Profiles; Auswahlliste

Durch Anklicken eines dieser Objekte wird das Element in das Feld übernommen. Mittels des *Cancel* Buttons kann von der Auswahlmaske, auch ohne eine Auswahl zu treffen, wieder in den Editor-Bereich zurückgesprungen werden.

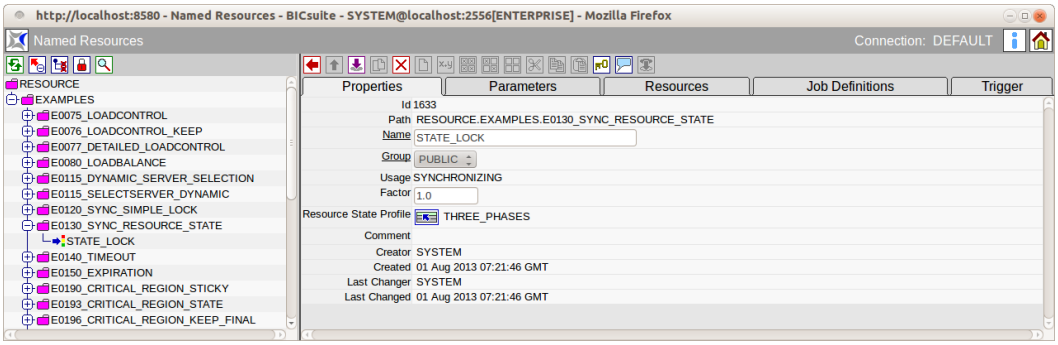


Abbildung 1.19: Auswahl eines Resource State Profiles; Resultat

1.6 Standard Listen Handling

In vielen Dialogen gibt es die Möglichkeit, nicht nur Werte in Felder einzugeben, sondern darüber hinaus noch Listen mit Verknüpfungen auf Zusatzobjekte zu erstellen. So wird z. B. innerhalb eines Resource State Profiles eine Liste von Resource States die zu diesem Profile gehören verwaltet. Ein Beispiel dazu:

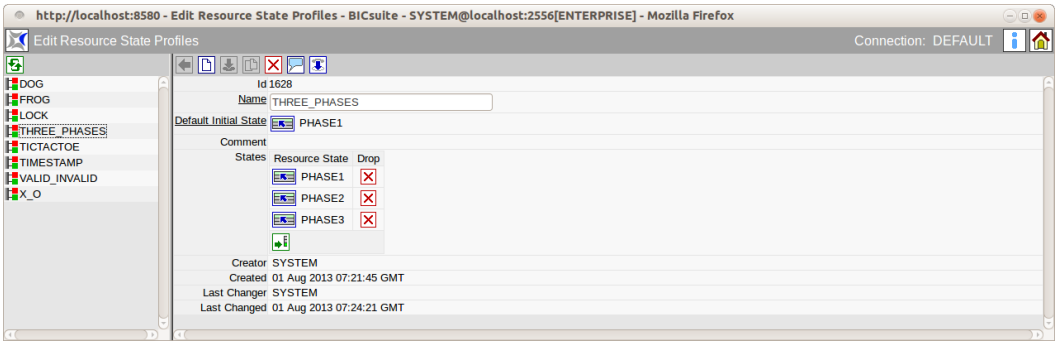


Abbildung 1.20: Beispiel Standard Listen Handling

Hier ist zu den normalen Feldern *Name* und *Default-Initial-State* noch eine Liste von States anzugeben.
Innerhalb einer solche Liste können nun die nachfolgenden Aktionen durchgeführt werden.

1.6.1 Hinzufügen einer Zeile


Hinzufügen

Mit dem *Hinzufügen*-Button kann eine neue Zeile zur aktuellen Liste hinzugefügt werden. Nach dem Betätigen des Buttons erscheint eine leere Zeile.

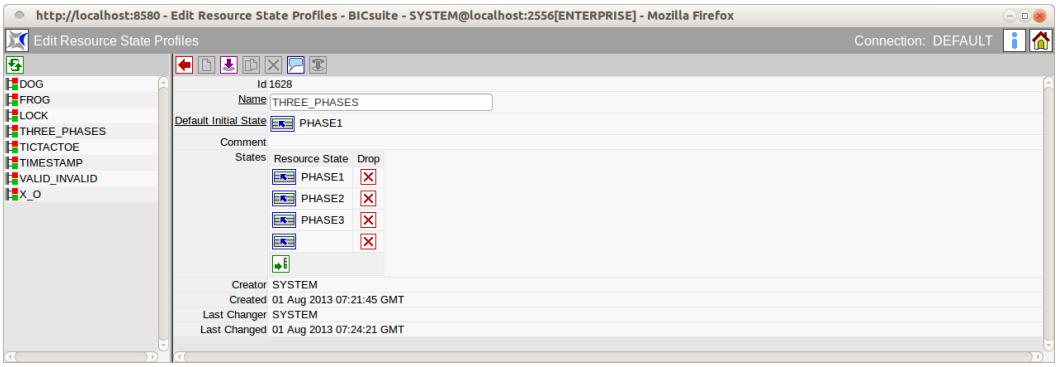


Abbildung 1.21: Listen Handling; Zufügen einer Zeile

In unserem Beispiel sieht das dann aus wie in Bild 1.21.
Anschließend kann über den Auswahlbutton ein Wert ausgewählt werden.

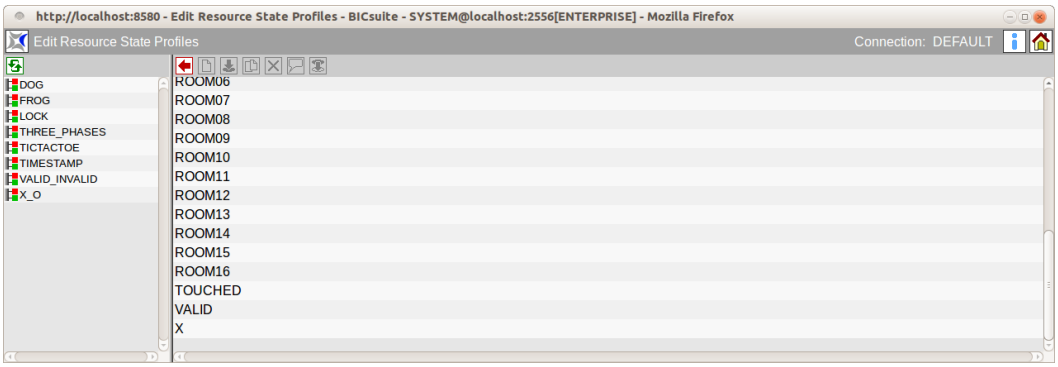


Abbildung 1.22: Listen Handling; Auswahl eines Wertes

In diesem Beispiel wurde der Wert ”Valid” ausgewählt.
Die Änderung muss nun mit Hilfe des Save Buttons gespeichert werden.
Die Möglichkeit der Auswahl eines abhängigen Objektes durch den *Auswahl*-Button, kann auch für die Änderung einer bestehenden Zeile durchgeführt werden. Hier wird dann die Verknüpfung zum alten Objekt gelöst und eine neue Verknüpfung zum gewählten Objekt aufgebaut.

1.6.2 Löschen einer Zeile

Drop (Löschen)
Mittels des *Drop* Buttons wird eine Zeile entfernt und die Verknüpfung vom aktuell im Editor geladenen Objekt und dem gewählten Objekt in der Zeile nach erfolgreicher Sicherheitsabfrage gelöscht. Im Beispiel wird durch Betätigen des *Drop*

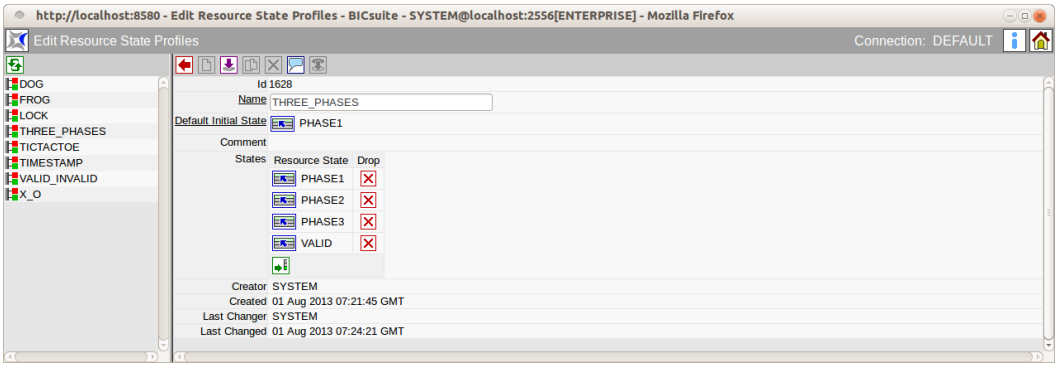


Abbildung 1.23: Listen Handling; Resultat

Buttons die Zeile ”Phase1” gelöscht.

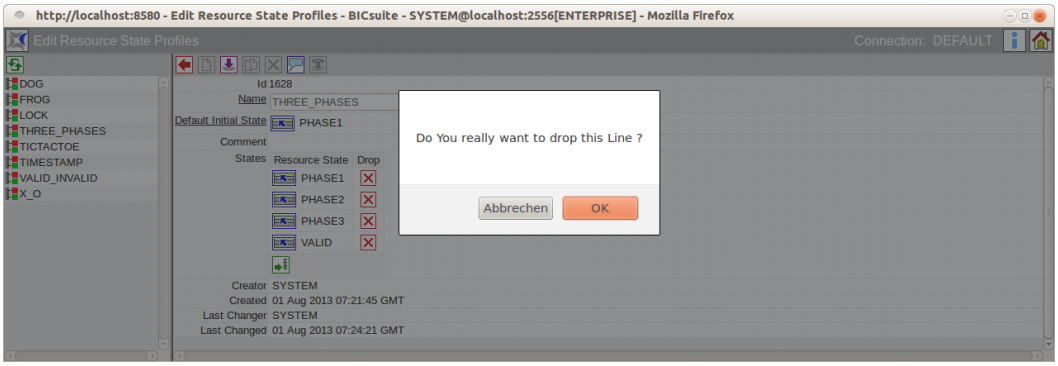


Abbildung 1.24: Listen Handling; Löschen einer Zeile

Die Zeile mit Phase1 ist nach Drücken des *OK* Buttons nicht mehr vorhanden. Das Objekt der Zeile (also der Resource State ”Phase1”) ist nach wie vor vorhanden, nur ist es nicht mehr diesem Profile zugeordnet.
Anschließend muss die Änderung mittels des *Speichern*-Buttons auf dem Server gespeichert werden.

1.6.3 Ändern von Zeilenwerten

In vielen Dialogen besteht eine Zeile nicht nur aus Zusatzeigenschaften zu anderen Objekten, sondern darüber hinaus sind zu dieser Verknüpfung auch noch Zusatzfelder zu pflegen. Diese werden als Feldleiste im Anschluss an das verknüpfte Objekt in der Liste angezeigt. Zum Beispiel ist im Dialog Footprint innerhalb einer Zeile nicht nur eine Named Resource auszuwählen, sondern darüber hinaus noch die weiteren Felder *Amount* und *Keep* zu pflegen.
Dieses geschieht analog zum Editieren von normalen Feldern.

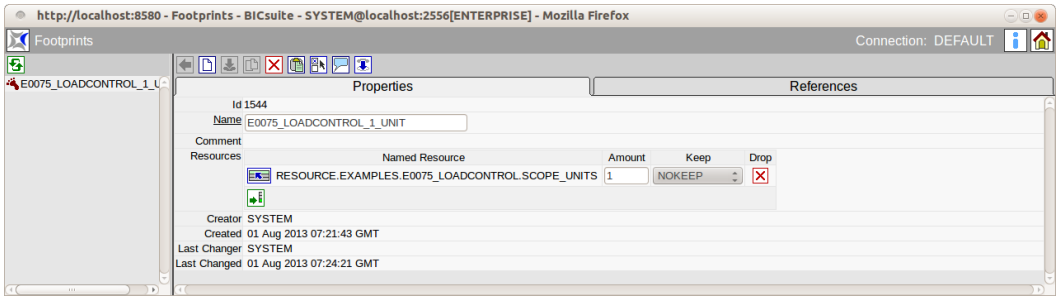


Abbildung 1.25: Listen Handling; Editieren

Anschließend müssen die Änderungen mittels des *Speichern*-Buttons gespeichert werden.

1.7 Copy und Paste und Zwischenablage

Die BICsuite!web-Oberfläche implementiert einen intelligenten Copy und Paste-Mechanismus und eine objektsensitive Zwischenablage. Hiermit ist es möglich, zwischen einzelnen Dialogen oder innerhalb eines Dialoges Objekte zu markieren, auszuschneiden, einzusetzen und zu löschen. Die Daten befinden sich, solange die Aktion nicht durchgeführt wurde, in der Zwischenablage des BICsuite!web-Oberflächen-Systems und können von geeigneten anderen Dialogen benutzt werden.

Der Begriff *o*bjektsensitiv beschreibt die Logik, die beim Einsetzen von Objekten aus der Zwischenablage durchgeführt wird. Vor dem Einfügen wird überprüft, ob die Objekte aus der Zwischenablage vom selben Typ sind, wie die benötigten Objekte des Dialoges in dem Eingefügt wird. Es ist also nicht möglich ein Objekt vom Typ A einzufügen, wenn der Dialog in dem eingefügt werden soll ein Objekt B erwartet. Gibt es keine Objekte vom Typ B in der Zwischenablage kann keine Einfügeoperation durchgeführt werden.

Dieser Mechanismus ist nicht zu verwechseln mit dem normalen Zwischenablagesystem des Windows-Betriebssystems. Die BICsuite Zwischenablage ist vollkommen unabhängig von der Windows-Zwischenablage und umgekehrt. Die Daten welche in der BICsuite Zwischenablage vorgehalten werden, stehen ausschließlich BICsuite Dialogen zur Verfügung und können nicht von anderen Windows-Applikationen benutzt oder ausgelesen werden.

Die Verwendung der Zwischenablage wird anhand zwei Beispiele in den folgenden Abschnitten erläutert.

1.7.1 Verschieben von Objekten in der Hierarchie

In einer Ordnerhierarchie kann man mit der Zwischenablage Objekte in eine andere Hierarchiestufe verschieben.
Beispiel:
Im Navigationsbildschirm des *Batches and Jobs-Dialoges* gibt es z. B. folgende Hierarchie:

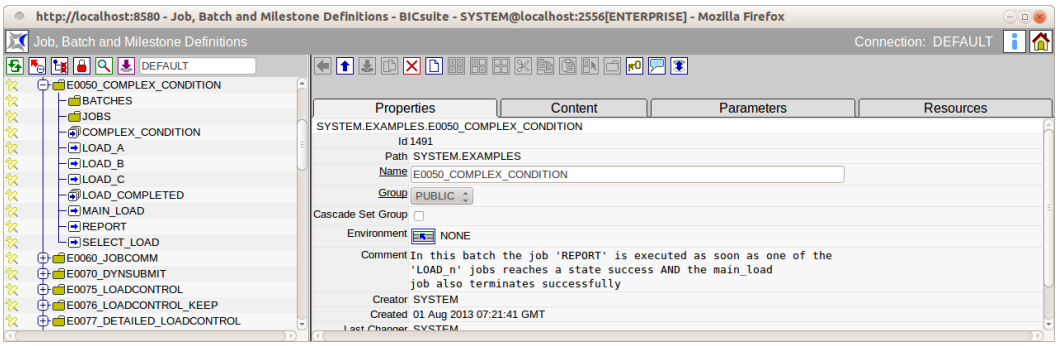


Abbildung 1.26: Objekte verschieben; Anfangszustand

Unterhalb des ”COMPLEX_CONDITION” Ordners gibt es verschiedene Jobs und Batches. Dieser Ordner soll nun geordnet werden.
Der Batch ”COMPLEX_CONDITION” soll in den Ordner ”BATCHES” verschoben werden und anschließend sollen die Jobs ”LOAD_A”, ”LOAD_B” und ”LOAD_C” in den Ordner ”JOBS” verschoben werden.
Mittels der Zwischenablage kann dies folgendermaßen durchgeführt werden:

1. Selektieren der zu verschiebenden Objekte

   Zuerst wird der Batch COMPLEX_CONDITION im Navigator angewählt. Auf dem ”Editor” Tab CONTENT erscheint die Liste aller Objekte welche im Ordner COMPLEX_CONDITION liegen.

   Anschließend wird der Batch durch Anklicken der Checkbox vor dem Namen und dem Icon markiert.

2. Ausschneiden der selektieren Objekte

   Mit dem *Cut* Button wird der selektierte Batch ausgeschnitten und verschwindet aus der Ansicht. Das Objekt wird nun in der Zwischenablage gespeichert. Der Dialog sieht nach dem Ausschneiden folgendermaßen aus:

   Es ist wichtig zu bemerken, dass das ausgeschnittene Objekt zwar nicht mehr sichtbar, jedoch immer noch in dem Folder vorhanden ist. Wird also jetzt das Fenster geschlossen, ist nichts passiert. Erst bei der späteren Paste-Operation wird die tatsächliche Verschiebung durchgeführt.

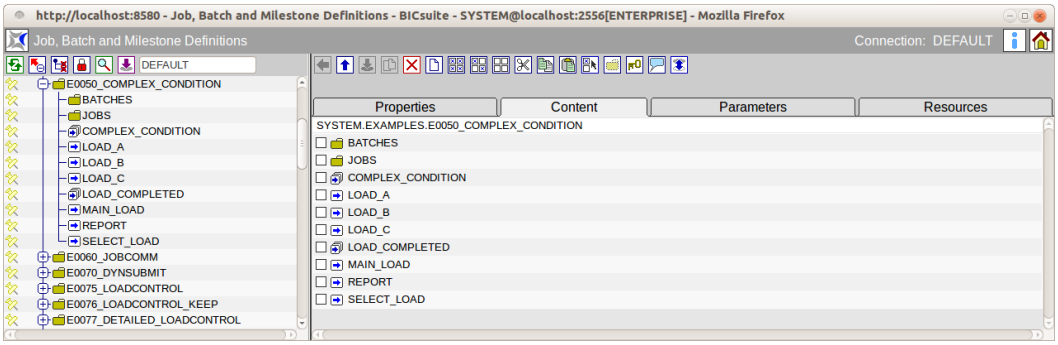


Abbildung 1.27: Objekte verschieben; Editor Content Tab

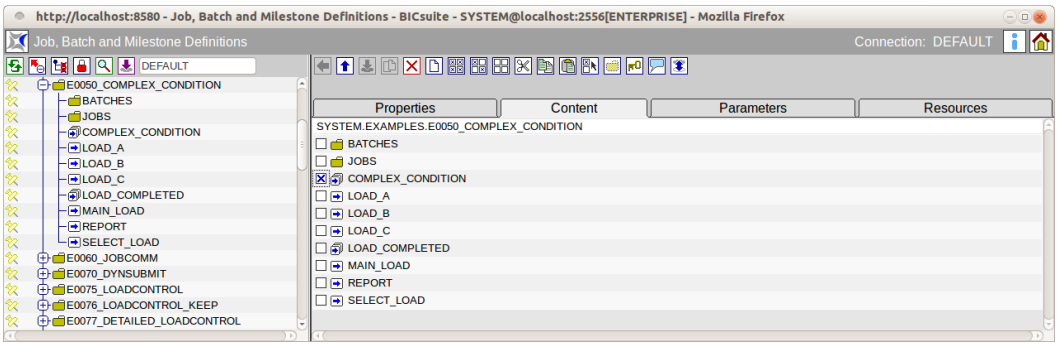


Abbildung 1.28: Objekte verschieben; Selektion

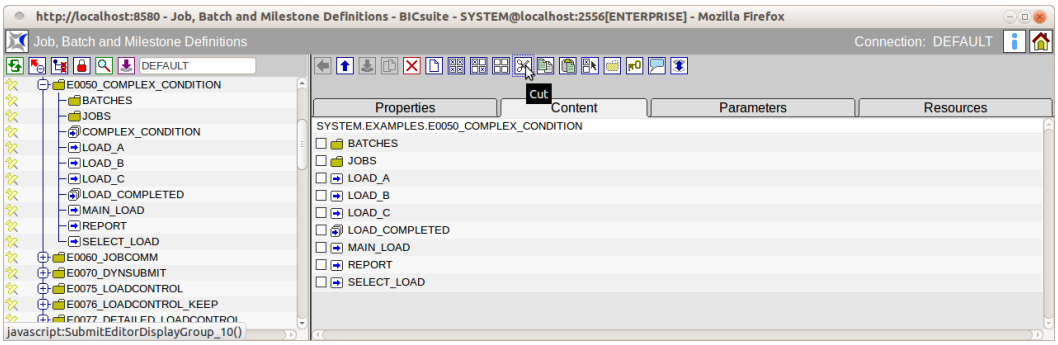


Abbildung 1.29: Objekte verschieben; Ausschneiden

3. Auswählen des Zielordners

   Nun werden der Zieldialog und der Zielordner für die Verschiebeoperation ausgewählt. Dies kann einfach durch das Anklicken des jeweiligen Ordners im Navigationsfenster des Zieldialoges geschehen. Da die Zwischenablage

fensterübergreifend funktioniert, kann sich der Zielordner jedoch auch in einem anderen Dialog befinden.

In unserem Beispiel wird nun der Ordner "BATCHES" gewählt und der Content Tab erscheint.

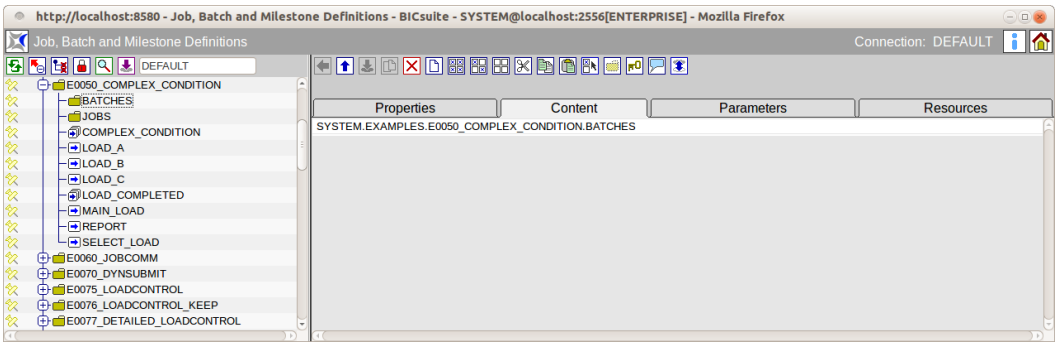


Abbildung 1.30: Objekte verschieben; Auswahl des Zielordners

4. Einfügen der selektierten Objekte

   Um die Verschiebung abzuschließen, wird das Objekt aus der Zwischenablage eingefügt. Dies geschieht mittels des *Paste* Buttons. Das Einfügen löst nun die Verschiebeaktion auf dem Server aus. Diese ist nun abgeschlossen und kann nicht mehr widerrufen werden. Das Ergebnis sieht so aus:

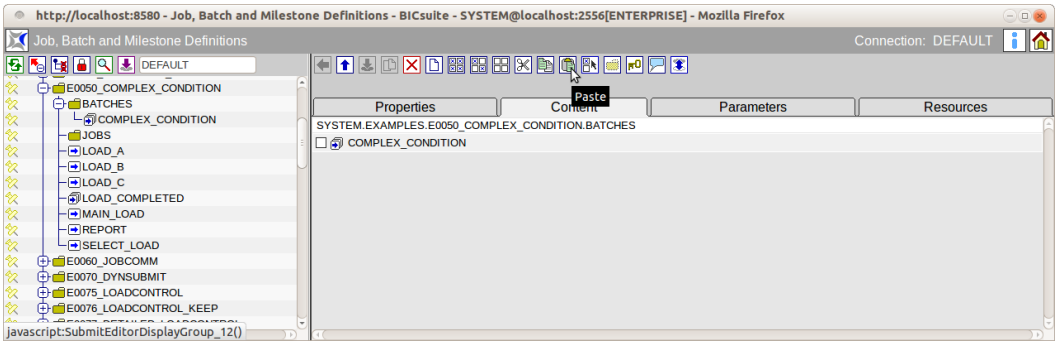


Abbildung 1.31: Objekte verschieben; Einfügen

Führt man die Aktion nun mit den oben genannten Jobs und dem Ordner "JOBS" aus, ergibt sich folgende Hierarchie:

1.7.2 Verknüpfung von Objekten

In einigen Dialogen werden Objekte mit anderen Objekten oder mit Objektlisten verknüpft. Ein Beispiel ist die Definition von Abhängigkeiten. Um so eine Liste zu

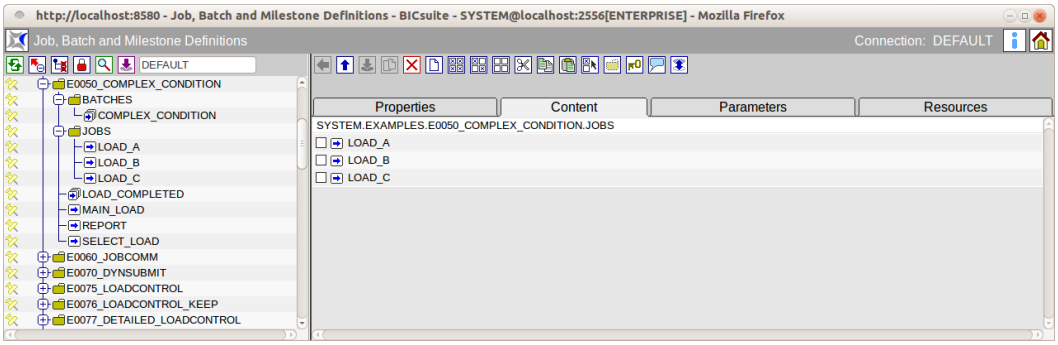


Abbildung 1.32: Objekte verschieben; Resultat

füllen, kann ebenfalls die Zwischenablage verwendet werden.
Um dies zu verdeutlichen und um den Ablauf einer solchen Aktion zu demonstrieren, soll die Definition von Abhängigkeiten eines Jobs durchgeführt werden.
Hier sehen sie den Job END_PROCESSING, der noch keine Abhängigkeiten hat:

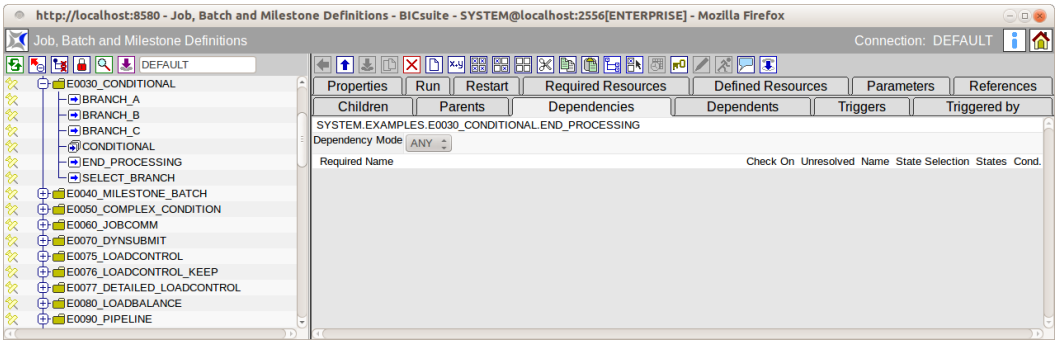


Abbildung 1.33: Verknüpfen von Objekten; Anfangssituation

Um die Ausführung dieses Jobs nun von anderen Verarbeitungsschritten abhängig zu machen, müssen an dieser Stelle andere Jobs eingefügt werden. Diese können vorher in anderen Dialogen markiert und in die Zwischenablage kopiert werden. Folgende Schritte sind durchzuführen:

1. Selektieren der Objekte

   Jobs können z. B. im Content Tab eines Folders markiert werden. Im nachfolgenden Dialog werden die drei Verarbeitungsschritte A, B und C ausgewählt.

2. Kopieren der Objekte in die Zwischenablage

   Durch Drücken des *Copy* Buttons werden die selektierten Jobs in die Zwischenablage kopiert.

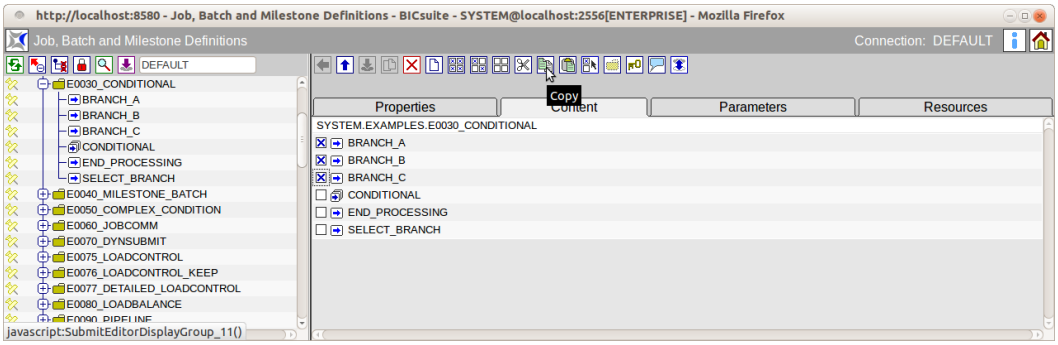


Abbildung 1.34: Verknüpfen von Objekten; Selektion

3. Einfügen der Objekte

   Nun kann im Dependency Editor Fenster des Jobs "END_PROCESSING" der *Paste* Button gedrückt werden und die zwischengespeicherten Jobs werden zur Liste hinzugefügt. Die Liste sieht anschließend folgendermaßen aus:

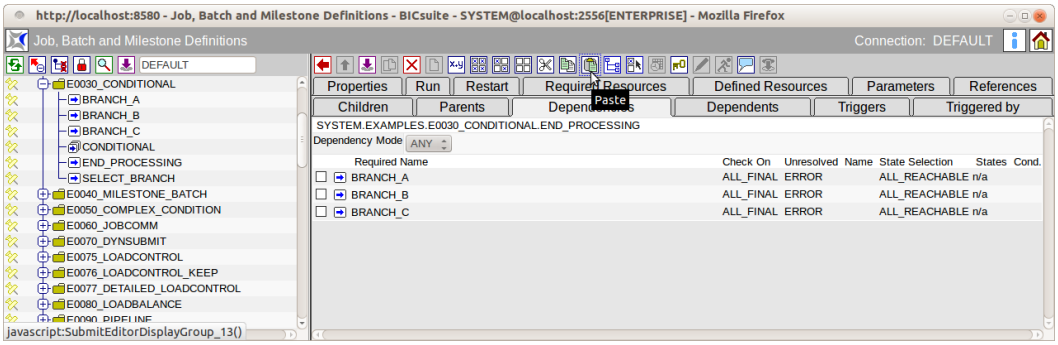


Abbildung 1.35: Verknüpfen von Objekten; Einfügen

4. Speichern der Änderung

   Im Gegensatz zum Verschieben von Objekten, bei dem kein zusätzliches Speichern mehr erfolgen muss, müssen Kopieraktionen in ein Editor Fenster mit dem *Speichern*-Button gespeichert werden.

   Achtung: Der Inhalt der Zwischenablage bleibt weiterhin erhalten. Sollen die selektierten Jobs auch in anderen Dialogen hinzugefügt werden, so ist es nur nötig das entsprechende Ziel auszuwählen und anschließend den *Paste* Button zu betätigen.

1.8 Grafik-Legende

In der folgenden Dokumentation werden verschiedene Grafiken zur Illustration von Zusammenhängen verwendet.

1.8.1 Darstellung der BICsuite Objekte

In den Grafiken werden folgende Symbole und deren Bedeutungen für BICsuite Objekte verwendet:

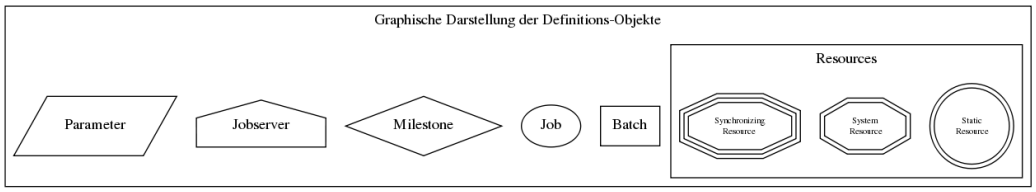


Abbildung 1.36: Legende graphische Abbildung BICsuite Objekte

1.8.2 Darstellung der Parent-Child-Beziehungen in Abläufe

In den Grafiken wird folgende Darstellung der Beziehung zwischen Parent und Child verwendet:

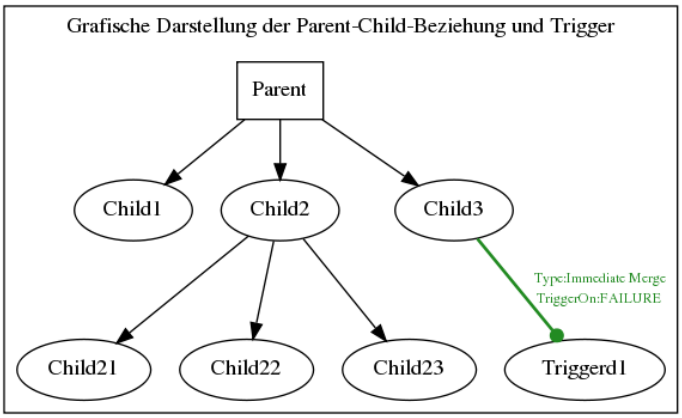


Abbildung 1.37: Legende graphische Darstellung Parent-Child-Beziehung

Die Darstellung geht normalerweise von oben nach unten, vom obersten Parent zu den jeweiligen Children. Die Richtung der Pfeile zeigt immer vom Parent zu den Children.

Mehr zu Parent-Child-Beziehungen zwischen Ablaufobjekten finden Sie im Kapitel Batches und Jobs.

1.8.3 Darstellung der Abhängigkeiten zwischen Scheduling Entities

In den Grafiken wird folgende Darstellung der Abhängigkeit zwischen einzelnen Scheduling Entities verwendet:

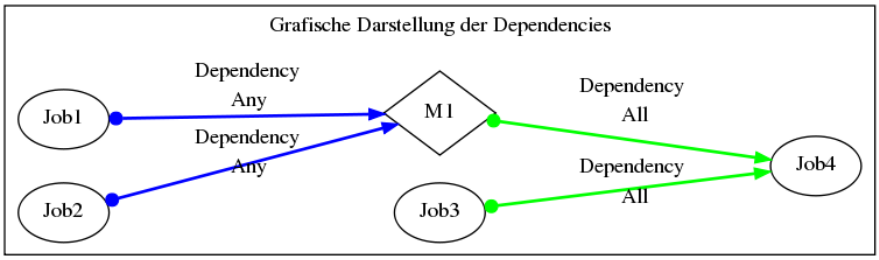


Abbildung 1.38: Legende graphische Darstellung Abhängigkeitsbeziehung

Die Darstellung der Abhängigkeiten erfolgt von rechts nach links. Das heißt, der erste Job wird ganz links angezeigt, anschließend kommt der zeitlich nächste usw.

Die Richtung der Pfeile geht immer vom benötigten Scheduling Entity (required Entity) zum abhängigen Scheduling Entity (dependent Entity) an der Pfeilspitze. Sie stellt demnach die zeitliche Abfolge der Scheduling Entities dar. Also wenn A→B→C dargestellt wird, wird erst A ausgeführt anschließend darf erst B und dann C ausgeführt werden. A ist dabei das benötigte Entity von B und B ist das benötigte Entity von C. B ist demnach von A abhängig. C ist von B (und implizit dann auch von A) abhängig.

Die Farbe der Pfeile entspricht der Verknüpfung mehrerer Abhängigkeiten.

Ist die Farbe Grün, so sind alle benötigten Tasks mittels ”UND” verknüpft (Dependency ALL).

Ist die Farbe Blau, so sind alle benötigten Tasks mittels ”ODER” verknüpft (Dependency ANY).

Ist ein Ablaufobjekt nur von genau einem anderen Ablaufobjekt abhängig, wird der Pfeil grün dargestellt, da in diesem Fall die Verknüpfungen UND und ODER gleichwertig sind.

Mehr zu Abhängigkeitsbeziehungen zwischen Scheduling Entities finden Sie im Kapitel Batches und Jobs.

1.9 Rechte

1.9.1 Kurzbeschreibung

Beim BICsuite Scheduling System handelt es sich um ein System, mit dem man Programme auf andere Rechner starten kann. Daher ist es wichtig, dass die Zugriffsrechte detailliert festgelegt werden können.

	View	Edit	Drop	Use	Create
Folder	•	•	•		•
Job Definition	•	•	•		
Named Resource	•	•	•		•
Scope	•	•	•		•
Jobserver	•	•	•		•
Resource	•	•	•		
Environment	•			•	
Group					

	Submit	Monitor	Operate	Resource	Execute
Folder	•				
Job Definition	•	•	•		
Named Resource				•	
Scope				•	•
Jobserver				•	•
Resource					
Environment					
Group		•	•		

1.9.2 Ausführliche Beschreibung

Für jedes Objekt sind Zugriffsrechte definiert. Die Zugriffsrechte variieren von Objekt zu Objekt.
Die Zugriffsrechte haben folgende Bedeutung:

View Mit dem *View*-Recht ist die Definition des Objektes sichtbar.

Edit Mit dem *Edit*-Recht ist es erlaubt das Objekt zu ändern.

Drop Mit dem *Drop*-Recht ist es erlaubt das Objekt zu löschen.

Use *Use* bezieht sich nur auf Environments. Das *Use*-Recht sagt aus, dass das betreffende Environment verwendet werden darf.

Create Mit dem *Create*-Recht ist es erlaubt in der betreffenden Umgebung Objekte anzulegen.

Submit Mit dem *Submit*-Recht hat man das Recht den Job zu submitten.

Monitor Mit dem *Monitor*-Recht kann man das betreffenden Submitted Entity sehen. Das *Monitor*-Recht entspricht dem *View*-Recht.

Operate Mit dem *Operate*-Recht ist es erlaubt Änderungen an dem Submitted Entity durchzuführen.

Resource

- Named Resource: Man darf Instanzen von dieser Named Resource erzeugen.
- Scope: Im Scope kann man Ressourcen anlegen, dazu braucht man das *Resource*-Recht sowohl für die Named Resource als auch für den Scope.

Execute Das *Execute*-Recht bestimmt ob man auf dem Jobserver Jobs ausführen darf.



Abbildung 1.39: Beispiel der Rechtevergabe

Ab BICsuite Version 2.5.1 können Rechte in Objekthierarchien vererbt werden. Die bis 2.5.0 bestehenden Möglichkeit, Rechte rekursiv zu materialisieren, wird durch die neuen, wesentlich komfortableren 'Inherit Privileges' ersetzt.
Der Grant Dialog für hierarchische Objekttypen wie (Folder, Scope, ...) enthält für alle nicht Wurzelobjekte eine gelb dargestellte Zeile mit 'Inherit Grants from Parent', in welcher definiert werden kann, welche Rechte vom übergeordneten Objekt geerbt werden sollen.
Bei der Neuanlage von Objekten werden per Default alle Rechte als 'Inherit Privileges' gesetzt. Damit gelten, falls dies im Grant Dialog nicht verändert wird, die Rechte von Gruppen auf das übergeordnete Objekt auch für das untergeordnete Objekt.
Aus diesem Grunde ist es möglich, ein SUBMIT Recht auf ein FOLDER Objekt zu vergeben. Das SUBMIT Recht ist für Objekte vom Typ FOLDER zwar bedeutungslos, durch die Möglichkeit der Vererbung von Rechten kann aber ein, dem FOLDER untergeordnetes JOB oder BATCH Objekt, dieses SUBMIT Recht erben.

Der Grant Dialog zeigt (mindestens) eine Zeile für jede Gruppe, welche direkte oder geerbte Rechte auf ein Objekt hat, ausgenommen der Gruppe, welche Eigentümer des Objektes ist. Direkte Rechte werden durch ein schwarzes, geerbte Rechte durch ein graues Kreuz in der Checkbox angezeigt.

Werden Rechte geerbt, so zeigt der Grant Dialog für jede Gruppe in übergeordneten Zeilen an, welche Rechte von welchen übergeordneten Objekten geerbt wurden (Origin) und wem diese gehören (Origin Owner). Die Checkboxen dieser Zeilen können nicht verändert werden und die Hintergrundfarbe wird grau oder grün falls das Objekte der jeweiligen Gruppe gehört (Group = Origin Owner) dargestellt. Da diese grün dargestellten Zeilen der jeweiligen Gruppe gehören hat diese implizit alle Rechte auf dieses Objekt. Daher gibt es für solche Zeilen auch nie ein übergeordnetes Objekt aus dem Rechte geerbt werden. In übergeordneten Zeilen wird nicht angezeigt, welche Rechte die Gruppe auf dieses Objekt hat, sondern welche Rechte das Objekt für das der Grant Dialog geöffnet wurde, vom übergeordneten Objekt erbt.

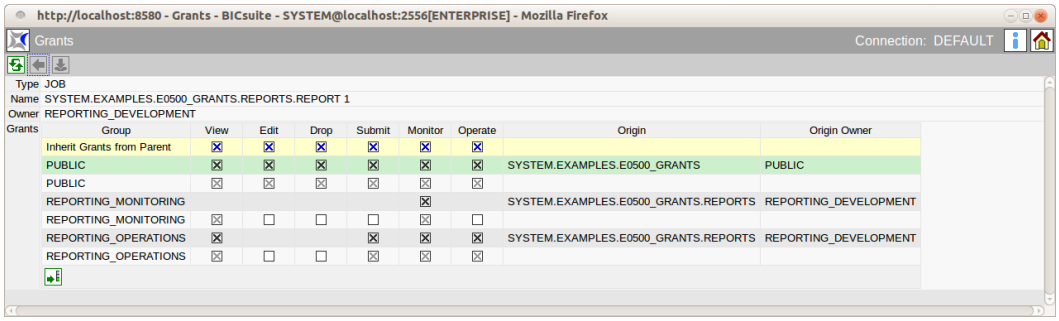


Abbildung 1.40: Darstellung der kumulierten Rechten

1.9.3 Systemprivilegien

Zusätzlich zu den Objektprivilegien gibt es auch Systemprivilegien. Diese Systemprivilegien dienen der Entlastung des Administrators und ermöglichen ein Delegieren der Pflege von Systemobjekten wie etwa Exit State Definitions.

Die Systemprivilegien werden, wie andere Privilegien auch, an Gruppen vergeben. Da sich diese Privilegien nicht auf einzelnen Objekten beziehen, erfolgt die Rechtevergabe im Gruppendialog, wie beschrieben im Kapitel groups.

Folgende Systemprivilegien sind vorgesehen:

Objekttyp	Bemerkung
Environment	Die Möglichkeit Environments zu ändern bzw. an zu legen kann dazu führen, dass der Benutzer auf beliebigen Jobservern Jobs ausführen kann. Daher sollte dieses Privileg nur mit Sorgfalt vergeben werden.
Exit State Definition	
Exit State Mapping	
Exit State Profile	
Exit State Translation	
Footprint	
Group	Dieses Privileg ist nur gültig für Gruppen in denen der Rechteinhaber selbst Mitglied ist. Damit ist gewährleistet, dass die Rechte eines Benutzers sich nicht expandieren können.
Resource State Definition	
Resource State Mapping	
Resource State Profile	
Select	Ermöglicht das Ausführen von SQL Select Anweisungen über den BICsuite!Server. Da dies einen lesenden Zugriff auf das gesamte BICsuite Repository ermöglicht, sollte dieses Privileg nur mit Sorgfalt vergeben werden.
System	kill session, stop server, alter server
User	Diese Privileg berechtigt dazu Benutzer im System zu verwalten. Allerdings sind die Möglichkeiten dazu eingeschränkt, um zu verhindern, dass ADMIN Rechte erlangt werden.

1.10 Comments

Für die meisten Objekte ist das Speichern von Comments möglich.

 Comment

Mittels des *Comment* Buttons kommt man in die Maske *Edit Comments*. Hier kann ein beliebiger Kommentar bzgl. des Objektes eingetragen werden. Es wird außer dem Kommentareintrag keine weitere Aktion am Objekt durchgeführt.

Alternativ zu einem Kommentar kann ein Link auf eine (Intranet)Seite erfasst werden. Dazu wählt man als Type "URL" und schreibt den URL ins Kommentarfeld. Die Links sind auch in der BICsuite!Web Oberfläche aktiviert. Auf diese Weise können beliebig komplexe Dokumente mit den BICsuite Objekten verknüpft werden.

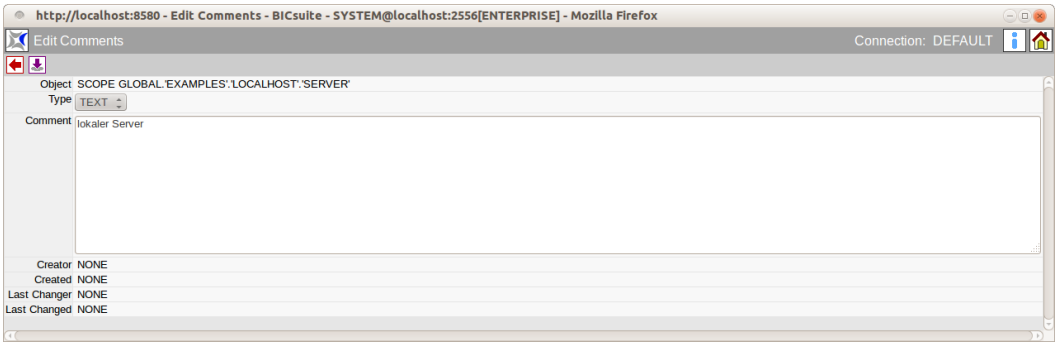


Abbildung 1.41: Erfassen von Kommentare

1.11 Standardfelder

Für alle Objekte sind folgende Felder definiert:

Creator Der Name des Benutzers der das Objekt angelegt hat.

Created Datum und Uhrzeit an dem dieses Objekt angelegt wurde.

Last Changer Name des Benutzers der zuletzt eine Änderung an dem Objekt vorgenommen hat.

Last Changed Datum und Uhrzeit an dem dieses Objekt zuletzt geändert wurde.

2 Exit State Definitions

2.1 Bild

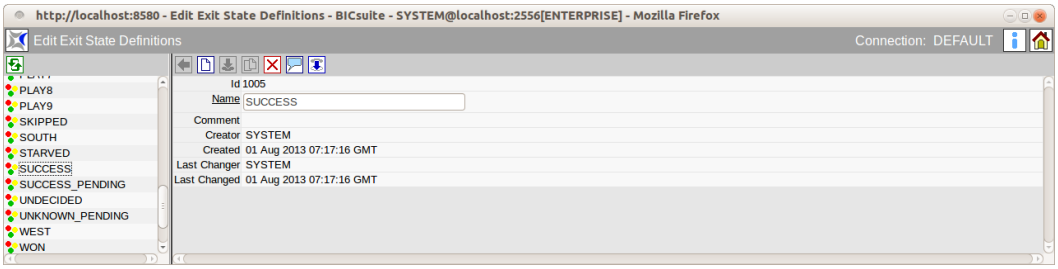


Abbildung 2.1: Exit State Definitions

2.2 Konzept

2.2.1 Kurzbeschreibung

Der Dialog Exit State Definitions dient zum Anlegen von logischen Namen für Endzustände eines Submitted Entity.

2.2.2 Ausführliche Beschreibung

Exit State Definitions sind logische Namen für Endzustände eines Submitted Entity. Diese logischen Namen besitzen keine weiteren Eigenschaften.

2.3 Editor

Dieser Tab dient der Pflege der Eigenschaften von Exit State Definitionen.
Exit State Definitionen dürfen nur von Benutzern, welche Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage exit state definition” Privileg sind, editiert werden.
Für alle anderen Benutzer sind alle Eingabefelder ”read only”.
Die Felder haben folgende Bedeutung:

Id Systemeindeutige Nummer zur Identifikation des Objektes.

Name Eindeutiger Name des Exit State Definition. Dieser Name kann frei gewählt werden.

3 Exit State Mappings

3.1 Bild

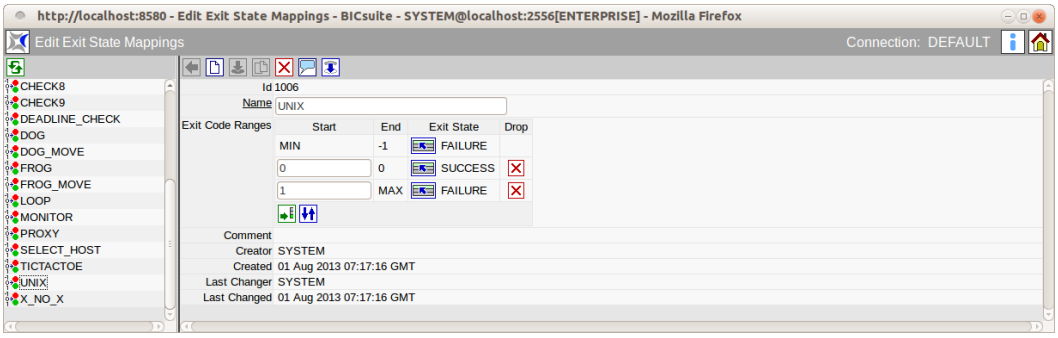


Abbildung 3.1: Exit State Mappings

3.2 Konzept

3.2.1 Kurzbeschreibung

Exit State Mappings dienen der Übersetzung von numerischen Exit Codes zu logischen Exit States.

3.2.2 Ausführliche Beschreibung

Mittels Exit State Mappings lassen sich numerische Wertebereiche von Exit States zu logischen Exit States zuordnen. Jedes Programm das ausgeführt wird, liefert am Ende einen numerischen Wert an das aufrufende Programm zurück. Unter UNIX-Systemen bedeutet dabei ein Wert von 0 normalerweise, dass das Programm ohne Fehler beendet wurde (SUCCESS). Werte ungleich 0 weisen auf einen Fehler hin (FAILURE). Es sind jedoch auch andere Werte mit anderen Bedeutungen denkbar. Um diese logische Bedeutung innerhalb von BICsuite besser zu verstehen, lassen sich mit Exit State Mappings Wertebereiche jeweils einem logischen Namen zuordnen. Die Werte reichen dabei von -2^{31} bis $2^{31} - 1$.

3.3 Editor

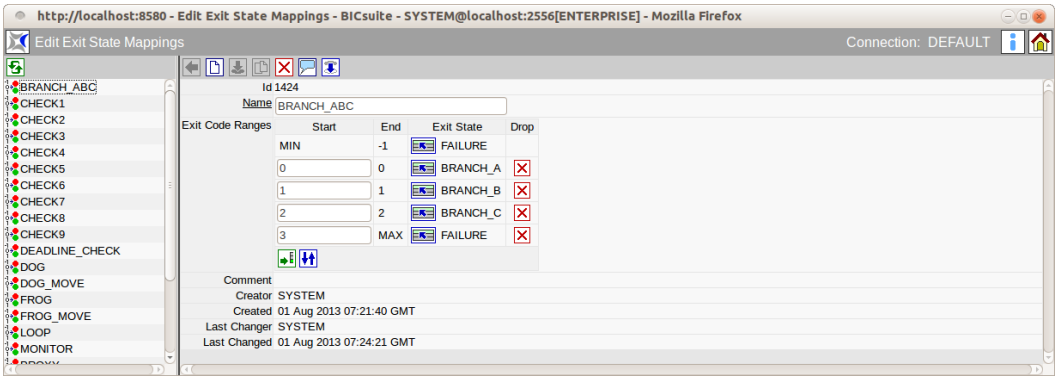


Abbildung 3.2: Exit State Mapping Editor

Im Editor werden die Zuordnungen der numerischen Werte zu den Namen vorgenommen.
Exit State Mappings dürfen nur von Benutzern, welche Mitglied der Gruppe "ADMIN" oder einer anderen Gruppe mit dem "manage exit state mapping" Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder "read only".
Die Beschreibung aller Standard-Buttons finden Sie im Kapitel 1.4.3.1 Standard-Buttons.
Die Buttons der obigen Liste haben folgende Bedeutung:

 Neuen Wertebereich einfügen

Dieser Button erweitert die Tabelle der Exit Code Ranges um eine weitere Zeile. Die Zeile wird am Ende eingefügt und ermöglicht die Eingabe eines neuen Startwertes. Der neue Endwert wird beim Speichern oder Sortieren automatisch ermittelt.

 Liste neu sortieren

Eine unsortierte Liste, die durch das Hinzufügen neuer Intervalle entsteht, kann mit diesem Button in die richtige Reihenfolge gebracht werden.

 Exit State Auswahl

Über diesen Button kann ein vorher definierter Exit State aus einer Liste gewählt werden.

 Löschen eines Wertebereiches

Dieser Button löscht einen Wertebereich zusammen mit dem Exit State aus der Liste. Der Startwert des nachfolgenden Intervalls erhält den Startwert des gelöschten Intervalls, d. h. das nachfolgende wird automatisch vergrößert.
Die Felder haben folgende Bedeutung:

Id Systemeindeutige Nummer zur Identifikation des Objektes.

Name Eindeutiger Name des Exit State Mappings. Dieser Name kann frei gewählt werden.

Exit Code Ranges Die Zuordnung der jeweiligen Wertebereiche wird in einer Tabelle dargestellt. In jeder Zeile kann einem Bereich ein Exit State zugeordnet werden. Die erste Zeile enthält den Startwert -2^{31}, die letzte Zeile hat als Endwert $2^{31} - 1$.

Editor

4 Exit State Profiles

4.1 Bild

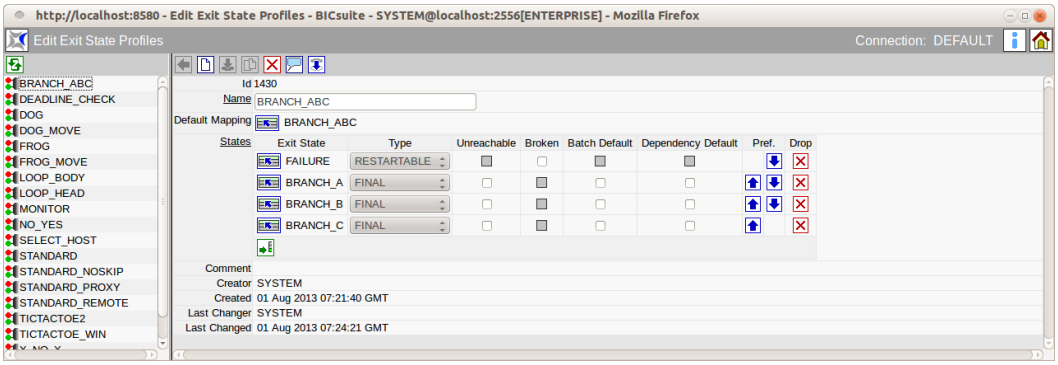


Abbildung 4.1: Exit State Profiles

4.2 Konzept

4.2.1 Kurzbeschreibung

Ein Exit State Profile beschreibt welche Exit States von einem Submitted Entity erreicht werden können. Außerdem gibt das Exit State Profile an, ob ein Submitted Entity seinen Exit State noch wechseln kann oder ob der Exit State endgültig (FINAL) ist.

4.2.2 Ausführliche Beschreibung

Jedes Submitted Entity, kann sich mit verschiedenen Exit States beenden. Die Menge aller gültigen Exit States eines Submitted Entities wird mit einem Exit State Profile beschrieben. In dieser Beschreibung wird auch die Präferenz eines Exit States festgelegt, die innerhalb einer hierarchischen Ablaufstruktur wichtig ist, um für ein übergeordnetes Element den korrekten Exit State zu bestimmen. Dabei gewinnt der Exit State mit der höchsten Präferenz.

4.3 Editor

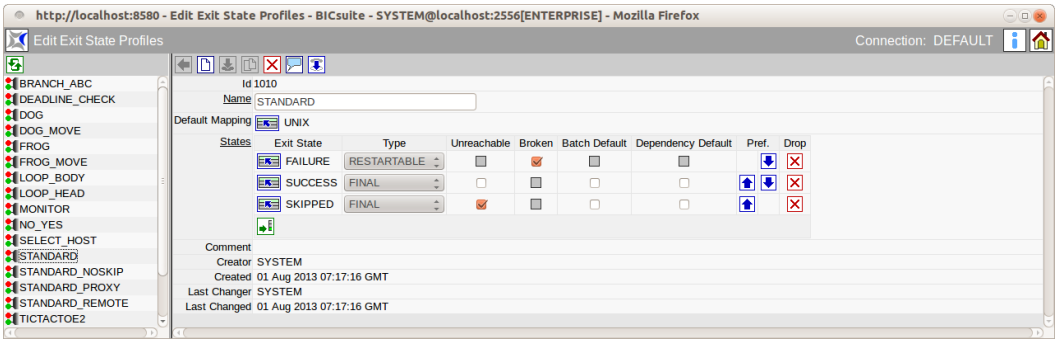


Abbildung 4.2: Exit State Profile Editor

Im Editor lassen sich die zu einem Profile gehörenden Exit States definieren und mit Eigenschaften (*Type*, *Unreachable*, *Broken*, *Preference*) versehen.
Exit State Profile dürfen nur von Benutzern, welche Mitglied der Gruppe "ADMIN" oder einer anderen Gruppe mit dem "manage exit state profile" Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder "read only".
Die obigen Felder haben folgende Bedeutung:

Name Über das Feld *Name* wird einem Exit State Profile ein eindeutiger Name zugewiesen.

Default Mapping Mit dem *Chooser* Button lässt sich aus einer Liste ein Exit State Mapping auswählen, das standardmäßig in Zusammenhang mit dem Exit State Profile verwendet wird, solange kein anderes Mapping in der Job Definition angegeben wird. Ein Exit State Mapping dient der Übersetzung von numerischen Exit Codes zu Exit States.
Für jeden dieser Exit States muss im Profile ein Eintrag vorhanden sein. Jedoch nicht alle Exit States eines Exit State Profiles müssen über das Mapping erreichbar sein.

Type Jedem Exit State kann einer von drei Typen zugewiesen werden: FINAL, RESTARTABLE und PENDING.
Hat ein Ablaufobjekt in einen Final State erreicht, kann der Exit State nicht mehr geändert werden. Das Ergebnis ist endgültig. Abhängigkeiten zwischen Jobs und Batches können nur auf solchen FINAL Exit States aufbauen.
Soll ein Ablaufobjekt nach seiner Ausführung nochmals gestartet werden können, z. B. weil ein Fehler aufgetreten war, so muss der Exit State vom Typ RESTARTABLE sein.

PENDING beschreibt einen State-Typ, der es ermöglicht den endgültigen State von außerhalb über das API zu setzen. Ein Ablaufobjekt, das in einem PENDING State steht, kann weder neu gestartet werden, noch werden dadurch irgendwelche Abhängigkeiten erfüllt.
Ein Beispiel für die Anwendung von PENDING:
Ein Job sendet einem Mitarbeiter eine E-Mail und bittet um Freigabe eines Ergebnisses. Nach dem Versenden der E-Mail beendet sich der Job mit einem PENDING State. Der Mitarbeiter kann nun den State manuell auf FINAL setzen. Alternativ kann ein Prozess, der die Antwort-E-Mail auswertet, die Statusänderung vornehmen. Erst nachdem der State FINAL ist, können abhängige Ablaufobjekte anlaufen.

Unreachable Maximal ein Exit State in der Liste darf als Unreachable markiert sein. Dieser Exit State wird gesetzt, wenn ein Job innerhalb eines Ablaufs nicht mehr zur Ausführung kommen kann, weil die Abhängigkeiten nicht mehr erfüllt werden können.
Der Unreachable State wird nicht angenommen, wenn ein oder mehrere Vorgänger gecancelled wurden. Gerade weil hier ein manueller Eingriff vorgenommen wurde, müssen auch die Folgen des Eingriffs manuell behandelt werden.
Der Unreachable State muss vom Typ ”Final” sein.

Broken Maximal ein Exit State in der Liste darf als Broken State markiert sein. Dieser State wird gesetzt, wenn ein Job aufgrund eines Fehlers in ein Error State versetzt wurde. Ein solcher Fehler trifft z. B. dann auf, wenn das Run Program nicht gestartet werden kann. Der Error State muss ein Restartable State sein. Durch Verwendung des Broken Flags kann mittels eines Triggers automatisch auf solche Fehlersituationen reagiert werden.

Batch Default Ein finaler Exit State in der Liste darf als Batch Default gekennzeichnet werden. Dieser State wird gesetzt wenn ein Batch oder Milestone mit diesem Profil keine Kinder hat, also auch keinen definierten Exit State hat. Ist kein Batch Default gesetzt so wird der State mit der niedrigsten Präferenz verwendet, der (falls vorhanden) nicht als Unreachable State gesetzt wurde.

Dependency Default Ein oder mehrere finale Exit States können als Dependency Default gekennzeichnet werden. Wird eine Abhängigkeit zwischen Batches bzw. Jobs oder Milestones mit State Selection DEFAULT angelegt, so erfüllen die als Dependency Default markierten Exit States es Profiles die Abhängigkeitsbedingung bzgl. des Exit States.

Preference  Über die *Preference* Buttons lässt sich die Präferenz der einzelnen Exit States festlegen. Dazu werden die Zeilen nach betätigen der But-

tons nach oben oder unten verschoben. Eine höhere Position bedeutet eine höhere Präferenz.
Bei der Ermittlung des resultierenden Exit State eines Submitted Entities, wird aus der Liste der Exit States der Children und des eigenen Exit States den Exit State mit höchster Präferenz gewählt. Dabei wird ein Exit State nur dann berücksichtigt, wenn der Exit State im Exit State Profile des Parents vorhanden ist oder der State mittels einer Exit State Translation nach einem Exit State des Parents übersetzt wird.
Beispiel: Ein Batch hat drei untergeordnete Child-Jobs. Der Batch soll genau dann einen Fehler anzeigen, wenn mindestens einer seiner Children einen Fehler meldet. Das bedeutet, der State FAILURE muss in der Reihenfolge vor SUCCESS kommen. In der Regel wird der SUCCESS State die niedrigste Priorität haben, Warnungen haben höhere Priorität und Fehler-States die höchste Priorität.
Wird ein leerer Batch (ein Batch ohne Children) FINAL, so gibt es keinen Exit State aus dem der FINAL Exit State des Batches ermittelt werden kann. In diesem Fall wird der Exit des Exit State Profiles verwendet, welcher die niedrigste Präferenz hat. Dies ist im allgemeinen SUCCESS. Dabei wird versucht einen FINAL State welcher nicht der Unreachable State ist zu verwenden falls ein solcher existiert.

5 Exit State Translations

5.1 Bild

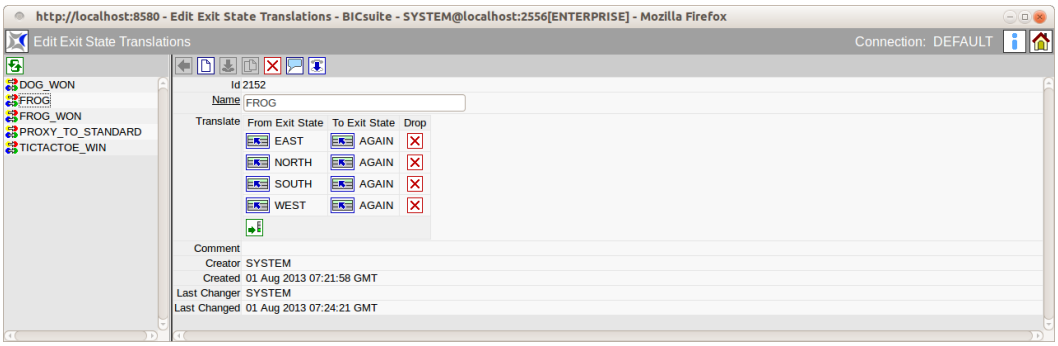


Abbildung 5.1: Exit State Translations

5.2 Konzept

5.2.1 Kurzbeschreibung

Exit State Translations werden benötigt, um den Exit State eines Childs zu einem Exit State des Parents zu übersetzen.

5.2.2 Ausführliche Beschreibung

Exit State Translations dienen der Übersetzung von Exit States eines Childs zu Exit States des Parents. Wenn keine Übersetzung vorhanden ist, wird die Identität genutzt oder der Exit State des Childs vom Parent ignoriert. Ist die Übersetzung vorhanden, müssen alle Exit States implizit oder explizit übersetzt werden können.

Verwendet zum Beispiel ein Child ein Exit State Profile mit den Exit State SUCCESS, WARNING, und FAILURE, und das Exit State Profile des Parents kennt bloß SUCCESS und FAILURE, könne die finalen Exit States, SUCCESS und WARNING nach dem Exit State SUCCESS des Parents übersetzt werden. Der restartable Exit State FAILURE wird nach Exit State FAILURE des Parents übersetzt.

5.3 Editor

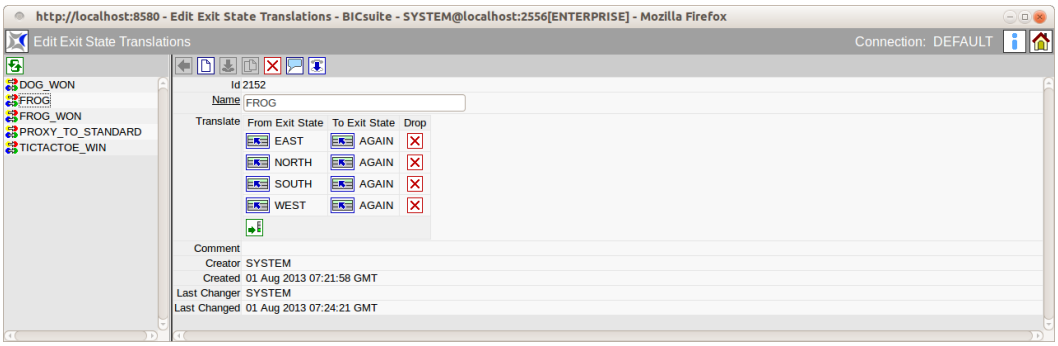


Abbildung 5.2: Exit State Translation Editor

Im Editor lassen sich die Exit State Translations definieren und anpassen.
Exit State Translations dürfen nur von Benutzern, welche Mitglied der Gruppe "ADMIN" oder einer anderen Gruppe mit dem "manage exit state translation" Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder "read only".
Die obigen Felder haben folgende Bedeutung:

Id Die Id enthält eine systemweit eindeutige Nummer.

Name Über das Feld *Name* wird der Exit State Translation ein eindeutiger Name zugewiesen.

Translate From/To Exit State In jeder Zeile wird angegeben, welcher State eines Childs nach welchem State des Parents übersetzt wird.

6 Resource State Definitions

6.1 Bild

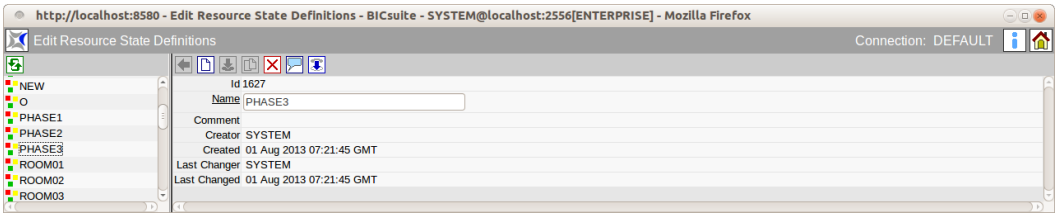


Abbildung 6.1: Resource State Definitions

6.2 Konzept

6.2.1 Kurzbeschreibung

Die Resource State Definitions dienen dem Anlegen von Namen für Resource States.

6.2.2 Ausführliche Beschreibung

Resource State Definitions sind logische Namen für Resource States. Jede Synchronizing Resource kann verschiedene States annehmen, die bei der Beendigung eines Jobs gesetzt werden können.

6.3 Editor

Dieser Tab dient der Pflege der Eigenschaften von Resource State Definitionen. Resource State Definitionen dürfen nur von Benutzern, welche Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage resource state definition” Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder ”read only”.
Die Felder haben folgende Bedeutung:

Id Systemeindeutige Nummer zur Identifikation des Objektes.

Name Eindeutiger Name der Resource State Definition. Dieser Name kann frei gewählt werden.

7 Resource State Profiles

7.1 Bild

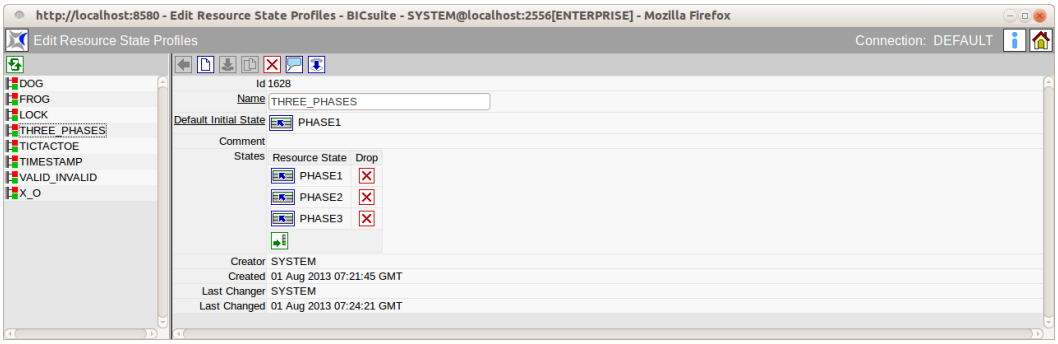


Abbildung 7.1: Resource State Profiles

7.2 Konzept

7.2.1 Kurzbeschreibung

Ein Resource State Profile beschreibt welche States eine Resource annehmen kann.

7.2.2 Ausführliche Beschreibung

Jede Synchronizing Resource kann verschiedene Resource States annehmen. Die Menge aller gültigen Resource States einer Resource wird mit einem Resource State Profile beschrieben. In dieser Beschreibung wird auch der initiale State einer Resource festgelegt. Der initiale Status muss dabei nicht unbedingt in der Liste der States vorkommen.

7.3 Editor

Im Editor lassen sich die zu einem Resource State Profile gehörenden Resource States definieren.
Resource State Profile dürfen nur von Benutzern, welche Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage resource state profile” Privileg

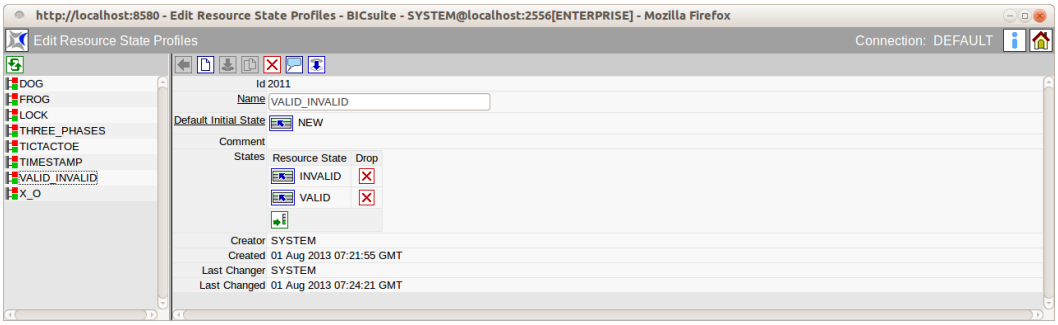


Abbildung 7.2: Resource State Profile Editor

sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder ”read only”.
Die obigen Felder haben folgende Bedeutung:

Name Über das Feld *Name* wird dem Resource State Profile ein eindeutiger Name zugewiesen.

Default Initial State Das Feld *Default Initial State* definiert den initialen State der Resource. Dieser Resource State muss nicht in der Liste gültiger Resource States vorhanden sein.

Resource State In der Tabelle *States* stehen in der Spalte *Resource State* die gültigen Resource States.

7.3.1 Beispiel

Eine Datenbanktabelle eines Datamarts wird als Synchronizing Resource mit State Modell im Scheduling System dargestellt. Das Zustandsdiagramm sieht wie im Bild 7.3 aus.
Die Definition eines geeigneten Resource State Profiles wird im obigen Screenshot dargestellt.

Editor

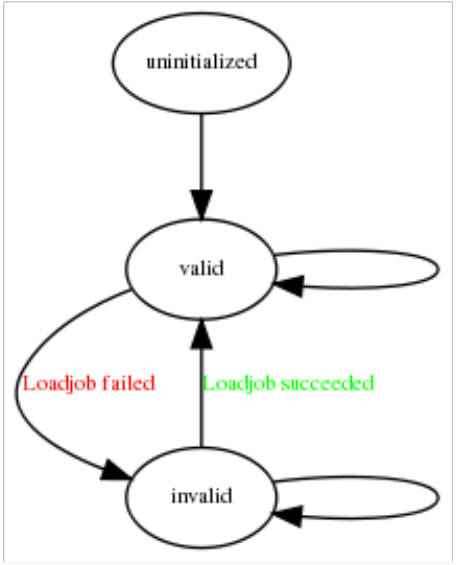


Abbildung 7.3: Zustandsdiagramm einer Resource

Editor

8 Resource State Mappings

8.1 Bild

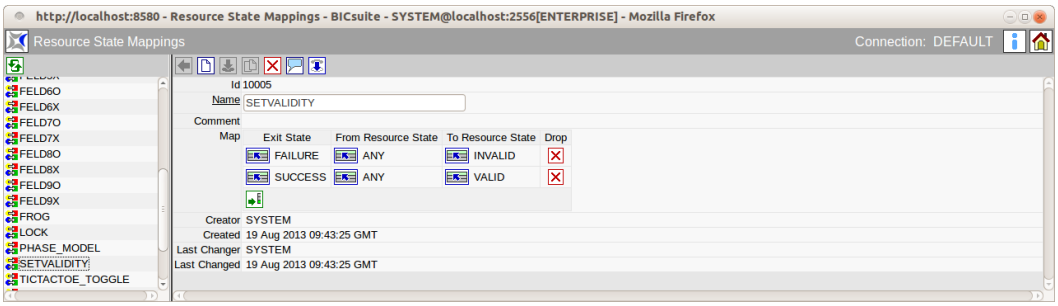


Abbildung 8.1: Resource State Mappings

8.2 Konzept

8.2.1 Kurzbeschreibung

Resource State Mappings beschreiben den Statuswechsel einer Resource bei bestimmten Exit States.

8.2.2 Ausführliche Beschreibung

Ressourcen können ihren Resource State nach der Beendigung eines Jobs in Abhängigkeit dessen Exit State verändern. Resource State Mappings beschreiben diese Statusübergänge. Jeweils ein Exit State und ein Resource State bestimmen den neuen State der Resource.
Beispiel: eine Resource ”TABELLE” kann den State ”VALID” oder ”INVALID” annehmen. Wird der zur Tabelle gehörende Ladeprozess erfolgreich beendet (SUCCESS), soll der Resource State ”VALID” gesetzt werden. Bei einer fehlerhaften Ausführung (FAILURE) soll die Tabelle als ”INVALID” gekennzeichnet werden. Die entsprechende Definition des Resource State Mappings würde wie im Bild 8.2 aussehen.
Existiert für die Kombination Exit State und From Resource State kein Mapping, wird der Status der Resource auch nicht verändert. So ist es nicht notwendig ein Mapping von SUCCESS/VALID nach VALID zu definieren.

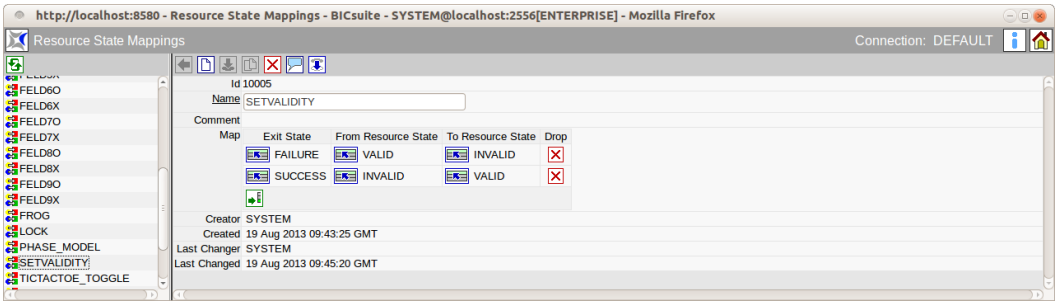


Abbildung 8.2: Resource State Mapping Beispiel

Soll der Statusübergang unabhängig von einem Resource State erfolgen, soll in der Zeile ”From Resource State”, der Wert ”ANY” gewählt werden.
Beim Umsetzen des Status einer Resource wird der Zeitpunkt des Umsetzens gespeichert. Bei Resource-Anforderungen kann auf dieser Zeit Bezug genommen werden, so dass auch das Setzen von ”VALID” nach ”VALID” sinnvoll sein kann.

8.3 Editor

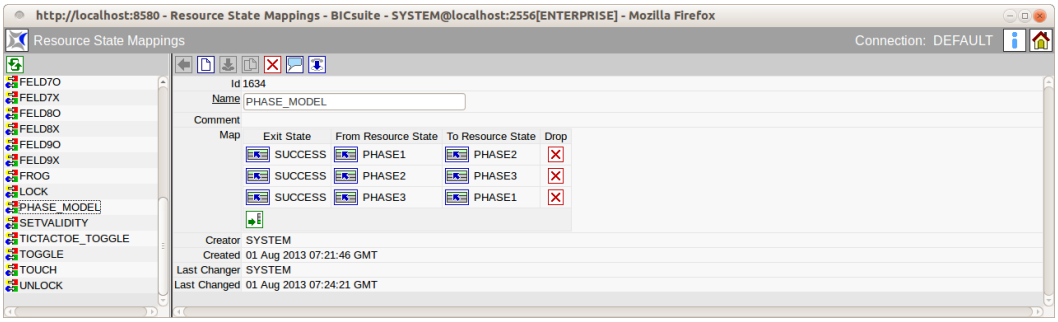


Abbildung 8.3: Resource State Mapping Editor

Im Editor werden die Zuordnungen der Exit States zu den Resource-Statusübergängen vorgenommen.
Resource State Mappings dürfen nur von Benutzern, welche Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage resource state mapping” Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder ”read only”.
Die Felder haben folgende Bedeutung:

Id Systemeindeutige Nummer zur Identifikation des Objektes.

Name Eindeutiger Name des Resource State Mappings. Dieser Name kann frei gewählt werden.

Exit State Der Exit State des Jobs steht im Feld *Exit State*.

From Resource State Im Feld *From Resource State* steht entweder ein bestimmter Resource State oder ANY. Falls der aktuelle State in Kombination mit dem Exit State explizit in der Tabelle genannt wird, wird beim Umsetzen des State diese Regel für die Bestimmung des neuen State herangezogen. Ansonsten wird die ANY-Regel, falls vorhanden, benutzt.

To Resource State Das Feld *To Resource State* ist der resultierende State der Resource.

Editor

9 Import/Export

9.1 Konzept

Mit dem *Import/Export* Button ist es möglich Objekte aus dem Scheduling System zu importieren und exportieren. Beim Exportieren wird ein Skript in der Kommandosprache des Systems erzeugt. Dieses Skript kann entweder im Browser angezeigt, oder sofort in eine Datei geschrieben werden.
Für eine ausführliche Beschreibung der Kommandosprache wird auf die ”BICsuite Server Command Reference” verwiesen.

9.2 Import/Export aus dem Main Desktop

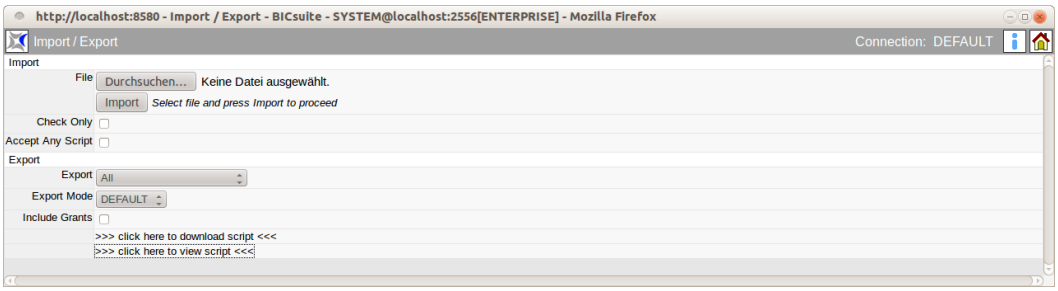


Abbildung 9.1: Import/Export aus dem Main Desktop

File Dieses Eingabefeld dient der Eingabe bzw. Auswahl der zu importierenden Datei. Da das Feld durch ein HTML Form Field vom Typ ’file’ abgebildet wird, ist das Look and Feel vom verwendeten Browser abhängig.

Import Durch Drücken des *Import* Buttons wird die ausgewählte Datei importiert.

Check Only Hiermit wird überprüft ob die Datei fehlerfrei eingelesen werden kann.

Accept Any Script Ist das Häkchen gesetzt, werden beliebige Skripte mit BIC-suite Kommandos akzeptiert. Ansonsten werden nur solche Skripte, die mittels der Export Funktionalität erzeugt wurden, akzeptiert.
Da beim Export einige Kommentare erzeugt werden, die zur Steuerung des Imports dienen, ist es im Allgemeinen nicht ratsam gedankenlos jedes Script zu akzeptieren.

Export Hier kann man einen Objekttyp, oder alle Objekttypen zum Export auswählen.

Export Mode Hierbei handelt es sich um die Art des Exports. In der Standard-Auslieferung des Systems, steht nur der Export Mode "Default" zur Verfügung. Wird die Export Funktion häufig und für unterschiedliche Zwecke genutzt, kann es sein, dass auch unterschiedliche Anforderungen an die Export Funktionalität gestellt werden. Diese Anforderungen können einfach und kundenspezifisch implementiert, und über den Export Mode auswählbar gemacht werden.

Include Grants Ist das Flag gesetzt, werden die Zugriffsrechte auf die zu exportierenden Objekte mit exportiert.

click here to download script Durch Anklicken dieses Links wird ein Export Script erzeugt und kann auf dem lokalen System gespeichert werden. Das Look and Feel des Dialogs zum Download des Scripts ist dabei vom verwendeten Browser abhängig.

click here to view script Durch Anklicken dieses Links wird ein Export Script erzeugt und im Browser angezeigt.

9.3 Import/Export Objektbezogen

Bei hierarchisch strukturierten Objekten wird das Objekt selbst und alle diesem Objekt untergeordneten Objekte exportiert. Beim Import wird alles unter dem Objekt, über welches der Import/Export Dialog aufgerufen wurde, importiert.
Zusätzlich zu den oben genannten Feldern sind noch folgende vorhanden:

Type Type gibt die Art des Objektes an.

Name Hierbei handelt es sich um den Namen des Objektes.

Include Schedules Dieses Feld wird nur für Objekte vom Typ FOLDER, BATCH und JOB angezeigt. Ist das Flag gesetzt, werden auch die für exportierte BATCH und JOB Objekte definierten Schedules mit exportiert.

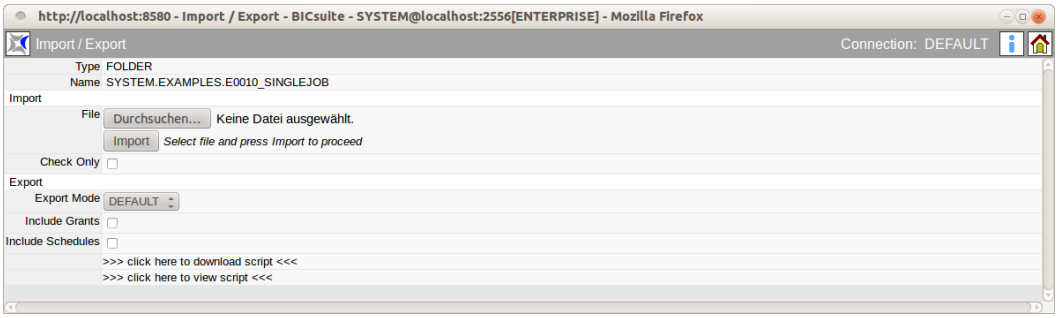


Abbildung 9.2: Objekbezogener Import/Export

10 Named Resources

10.1 Bild

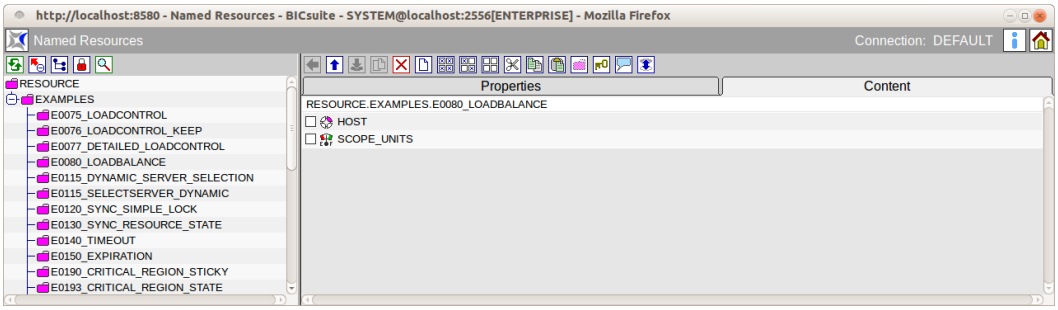


Abbildung 10.1: Named Resources

10.2 Konzept

10.2.1 Kurzbeschreibung

Dieser Bildschirm dient zum Erstellen und Verwalten von Named Resources. Eine Named Resource ist die Definition einer Klasse von Resources.

10.2.2 Ausführliche Beschreibung

Eine Named Resource definiert eine Klasse von Resources. Named Resources können innerhalb eines Scopes oder Folders bzw. eines Submitted Entities als Resources instanziiert werden.
Named Resources gehören einer Gruppe an. Die Benutzung (d.h. Instanziierung und Anforderung) sowie das Editieren oder Löschen der Named Resource ist in erster Linie Benutzern der Gruppe vorbehalten. Diese Rechte können anderen Gruppen zugeteilt werden.
Named Resources werden innerhalb einer Hierarchie von Kategorien gespeichert. Kategorien sind beliebig ineinander schachtelbar. Named Resources sind nicht schachtelbar. Die Kategorisierung dient ausschließlich der Verbesserung der Übersichtlichkeit bei der Verwaltung von Named Resources und hat darüber hinaus keinen Einfluss auf die Funktion des Systems.

10.3 Editor

Im Editor Menü werden Named Resources und Kategorien gepflegt. Wird im Navigator ein Eintrag gewählt, so erscheinen hier die Details dieser Named Resource oder Kategorie. Des weiteren können neue Named Resources und Kategorien erstellt werden.

10.3.1 Tab Properties bei Kategorien

Dieser Tab dient der Pflege von Kategorieeigenschaften. Er sieht folgendermaßen aus:

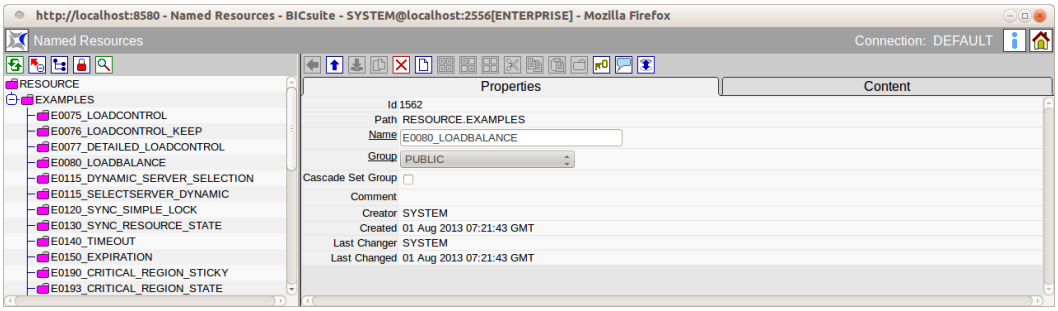


Abbildung 10.2: Kategorien; Properties Tab

Der Properties-Tab für eine Kategorie und Named Resources hat folgende Felder:

ID Die ID dient zum eindeutigen Identifizieren der Kategorie oder Named Resource Definition. Die ID wird automatisch vom System vergeben und ist systemweit eindeutig.

Path (Pfad) Der Path stellt den kompletten Pfad (Hierarchie) des aktuellen Eintrages dar. Die einzelnen Hierarchiestufen werden mittels eines Punktes getrennt.

Name Der Name der Kategorie oder der Named Resource kann frei gewählt werden. Er muss jedoch innerhalb der übergeordneten Kategorie eindeutig sein.

Group Die Gruppe, der die Kategorie zugeordnet ist, kann aus der "Drop Down" Liste ausgewählt werden.

Cascade Set Group Wird dieses Feld markiert. wird die Gruppe in allen Named Resources und Kategorien unterhalb dieser Kategorie ebenfalls gesetzt.

Comment Das *Comment* dient einer näheren Erläuterung des Objektes.

10.3.2 Tab Content

Der Tab Content wird nur bei Kategorien angezeigt und sieht folgendermaßen aus:

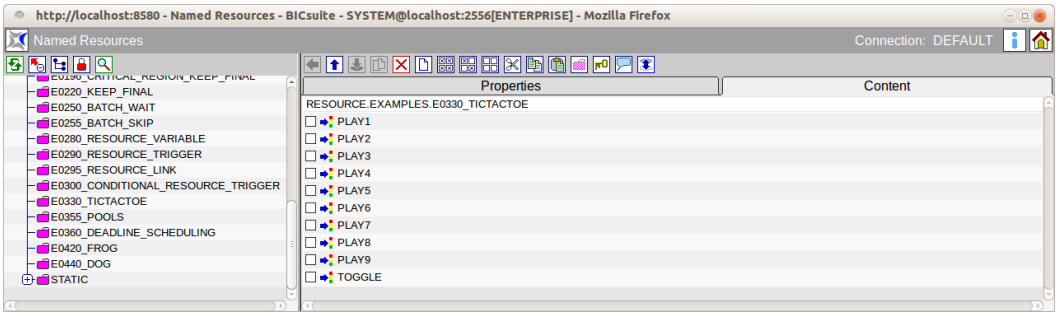


Abbildung 10.3: Kategorien; Content Tab

Der Tab enthält eine Liste von allen Named Resources, welche sich in der gewählten Kategorie befinden. Folgende Werte werden innerhalb der Liste angezeigt:

Name der Named Resource Hier erscheint der Name der Named Resource. In diesem Dialog kann der Name der Named Resource angewählt werden, was direkt zur Selektion und zur Anzeige der "Detail" Daten dieser Named Resource führt. In diesem Tab ist es möglich, mittels der normalen Cut, Copy und Paste-Methoden, Resources zu verschieben.

10.3.3 Tab Properties bei Named Resource Definitionen

Handelt es sich um eine Named Resource Definition sieht der Tab Properties folgendermaßen aus:

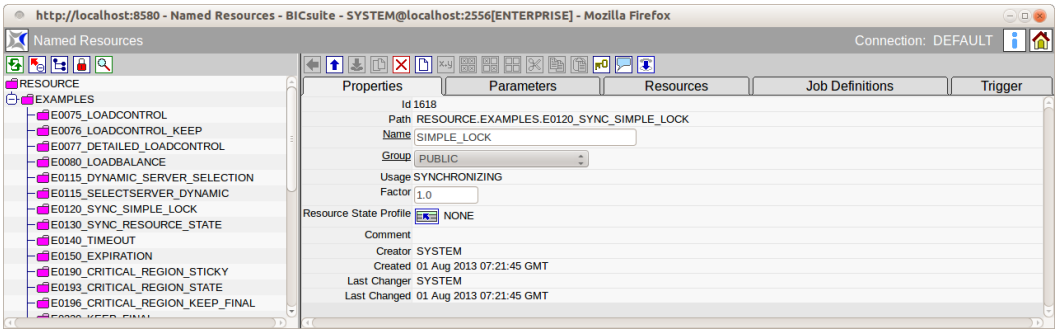


Abbildung 10.4: Named Resources; Properties Tab

Zusätzlich zu den o.g. Feldern, müssen bei der Definition einer Named Resource weitere Felder ausgefüllt werden.

Usage Die *Usage* gibt an, um welche Art einer Named Resource es sich handelt. Liste der Möglichkeiten für Usage Parameter:

1. *Category*

   Kategorien verhalten sich wie Folder und können benutzt werden um die Named Resources in eine Übersichtliche Hierarchie einzuordnen.

2. *Static*

   Statische Ressourcen beschreiben Ablaufumgebungen, die etwa durch installierte Software-Pakete, Benutzerumgebungen geboten werden. Eine Instanz dieser Named Resource muss vorhanden und verfügbar sein, damit ein Submitted Entity, welches diese Resource als Voraussetzung benötigt, starten kann.

3. *System*

   Die Named Resource stellt eine Resource dar, welche einen Systemparameter abbildet. Dies kann zum Beispiel eine CPU Einheit, eine Hauptspeichereinheit oder eine Einheit von Plattenplatz sein. Diese Resource muss beim Gebrauch zusätzlich quantifiziert werden, das heißt der Ersteller eines Jobs muss die Anforderungen, die dieses Objekt an System Resources stellt, qualifizieren und quantifizieren. Bei der Definition eines Jobs muss also zum Beispiel angegeben werden, dass 3 CPU Einheiten und 3 Einheiten à 512 MB Hauptspeicher und 10 Einheiten à 1 GB Plattenplatz benötigt werden.

4. *Synchronizing*

   Synchronizing Resources sind spezielle Ressourcen, die dazu dienen parallel laufenden Jobs zu synchronisieren. Ob eine Resource-Anforderung erfüllbar ist und die Resource belegbar ist, kann dabei vom gegenwärtigen Status der Resource, der letzten Status-Update-Zeit und der konkurrierenden Allocation der Resource durch andere Ablaufobjekte abhängig gemacht werden. Wie bei System Resources ist eine Quantifizierung der Anforderung möglich.

5. *Pool*

   Named Resources vom Typ *Pool* werden benutzt um sogenannte Resource Pools anzulegen. Diese Pools bieten die Möglichkeit die Verteilung von Amounts für System Resources zentral und flexibel zu regeln.

Resource State Profile Einer Synchronizing Resource kann ein Resource State Profile zugeordnet werden, welcher den Status den die Synchronizing Resource annehmen kann definiert. Wenn kein Resource State Profile angegeben ist, können Status- und Zeitstempelbezogene Anforderungen nicht definiert werden. Weitere Informationen finden Sie in den Kapiteln 6 und 7.

10.3.4 Tab Parameters

Im Tab Parameters können zusätzliche Informationen, die von Jobs oder Resource Trigger ausgewertet werden können, zu einer Resource gespeichert werden.

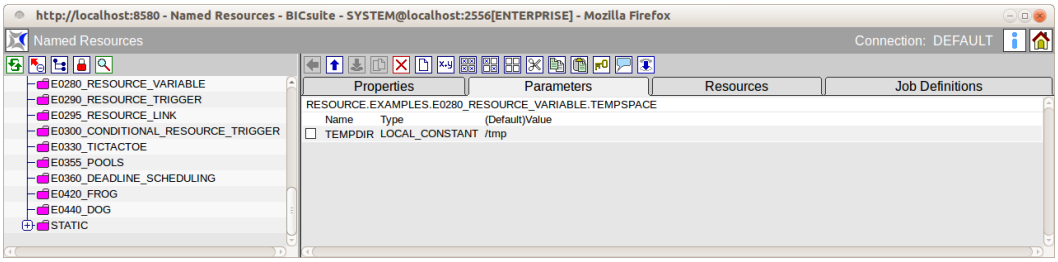


Abbildung 10.5: Named Resources; Parameter Tab

Der Tab Parameters sieht aus wie im Bild 10.5.
Durch anklicken des Parameter-Namens kommt man in den Tab ”Parameter Details”.

10.3.4.1 Tab Parameter Details

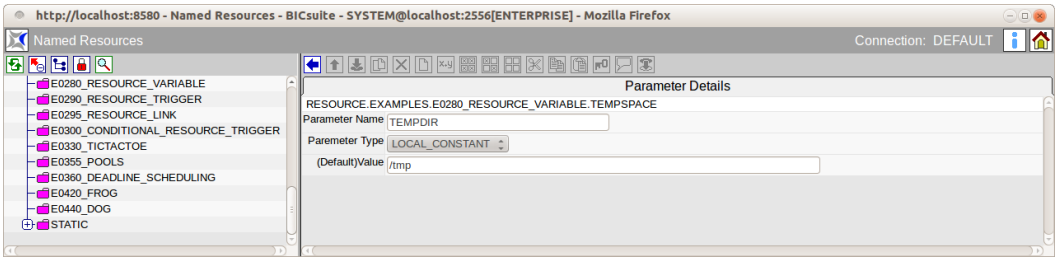


Abbildung 10.6: Named Resources; Parameter Details

Die Felder des Tabs ”Parameter Details” haben folgende Bedeutung:

Parameter Name Der Name des Parameters.

Parameter Type Beim Type handelt es sich um die Art des Parameters.
Es gibt folgende Möglichkeiten:

- Constant: Der Constant hat einen festen Wert und gilt für alle Ressourcen.
- Local Constant: Der Local Constant hat einen festen Wert, der von Resource zu Resource abweichen kann.

Default Value Beim Default Value unterscheiden wir zwischen Constants und Local Constants. Bei Constants ist er der Wert des Parameters und bei Local Constants der Default-Wert.

10.3.4.2 Standard Parameters

Ebenso wie es Standard Parameters für Jobs und Batches gibt, stehen auch für Ressourcen Standard Parameters zur Verfügung.
Um sie referenzieren zu können, muss, analog zu der Situation bei Batches und Jobs, ein kleiner Trick verwendet werden. Soll etwa den Inhalt der Standard Variable "STATE" für eine Verarbeitung genutzt werden, legt man eine weitere Variable, eine Konstante, mit als Wert "$STATE", an. Durch die rekursive Parameterauflösung erhält die Konstante den Wert des Parameters "STATE".
Folgende Standard Parameters stehen zur Verfügung:

STATE Der Status der Ressource. Für nicht synchronizing Ressourcen ist der Inhalt leer.

AMOUNT Die insgesamt zur Verfügung stehende Menge der Ressource. Für static Ressourcen ist der Wert leer.

FREE_AMOUNT Die zur Verfügung stehende freie Menge der Ressource. Für static Ressourcen ist der Wert leer.

REQUESTABLE_AMOUNT Die maximale Menge der Ressource, die beantragt werden darf. Für static Ressourcen ist der Wert leer.

10.3.5 Tab Resources

Wurde eine Named Resource in der Navigation selektiert, erscheinen Instanzen dieser in dem Tab Resources. Es zeigt alle Scopes, Folder, Submitted Entities, Scheduling Entities bzw. Jobserver die Instanzen dieser Named Resource anbieten an.
Der Tab Resources sieht folgendermaßen aus:

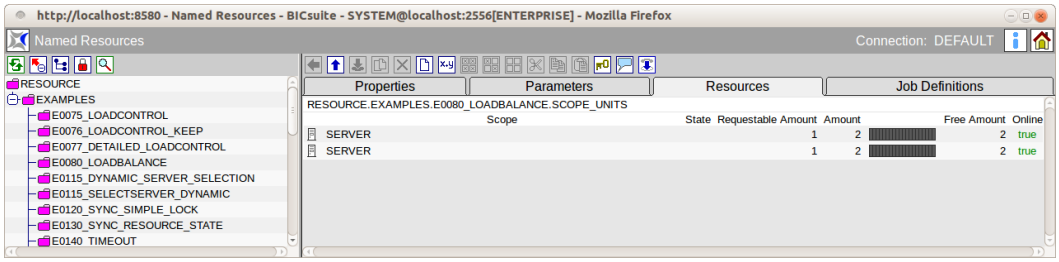


Abbildung 10.7: Named Resources; Instanzen

Die Instanziierungen der Named Resource werden durch folgende Felder Beschrieben:

Scope Hier erscheinen die Namen der Scopes, Submitted Entities, Scheduling Entities oder Folder, welche Instanzen (Resources) der jeweilige Named Resource anbieten.

State Besitzt die gewählte Named Resource ein Resource State Profile, wird hier der aktuelle Status der Resource angezeigt. Dies ist nur bei Synchronizing Resources möglich.

Requestable Amount Die Menge der Ressourcen die maximal von einem Job angefordert werden darf.

Free Amount Hier wird die Anzahl der freien Instanzen dieser Resource angezeigt. Ein Balken gibt den momentanen Belegungsgrad graphisch wieder.

Amount Die Amount ist die Anzahl der zur Verfügung stehenden Einheiten der Resource.

Online Der Verfügbarkeitsstatus der Resource.

10.3.6 Tab Job Definitions

Der Tab Job Definitions zeigt alle Job Definitions an, welche diese Named Resource anfordern. Dieser Tab dient lediglich zur Information, es können hier keine Änderungen vorgenommen werden.
Der Tab sieht folgendermaßen aus:

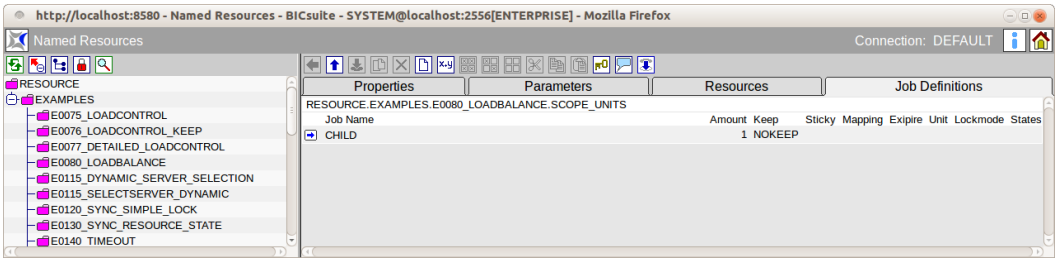


Abbildung 10.8: Named Resources; Job Definitions Tab

Die Liste der Jobs wird durch folgende Felder beschrieben:

Job Name Hier erscheinen die Namen der Submitted Entities, die die Named Resource benötigen.
Durch Anklicken des Namens wird ein Jobeditor-Fenster geöffnet.

Amount Die Menge der Resource, die von dem Job benötigt wird.

Keep Der Wert des Keep Parameters für die Resource-Anforderung des Jobs.

Sticky Der Wert des Sticky Flags für die Resource-Anforderung des Jobs.

Mapping Wurde bei der Resource-Anforderung ein Resource State Mapping angegeben, wird dies hier angezeigt.

Expire Der Expire-Wert gibt die maximale Zeit die seit dem letzten Statuswechsel der Resource vergangen sein darf um die Resource als belegbar zu betrachten. Nun hat eine negative Expiration zur Folge, dass eine Resource mindestens so ”alt” als angegeben sein muss.

Lockmode Der Lockmode beschreibt den Modus des Zugriffs auf diese Resource (exclusive, shared etc.).

States Zeigt alle Resource States an, die von diesem Job akzeptiert werden. Falls mehrere States für diesen Job akzeptabel sind, werden die einzelnen States durch Komma getrennt.

10.3.7 Tab Properties bei Pooled Resources

Named Resources vom Typ ”pool” werden benutzt um sogenannte Resource Pools anzulegen. Diese Pools bieten die Möglichkeit die Verteilung von Amounts für System Resources zentral und flexibel zu regeln.
Pooled Resources sind Resources die über den Pool verwaltet werden.
Die Beschreibung der einzelnen Felder siehe in ’Tab Properties bei Kategorien’ und ’Tab Properties bei Named Resource Definitionen’.
Die Beschreibung der einzelnen Felder siehe in ’Tab Resources’ bei den Named Resources.

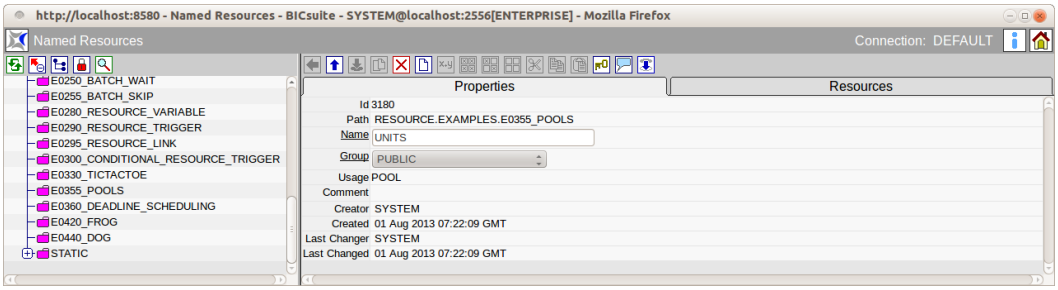


Abbildung 10.9: Named Resources; Pool Properties

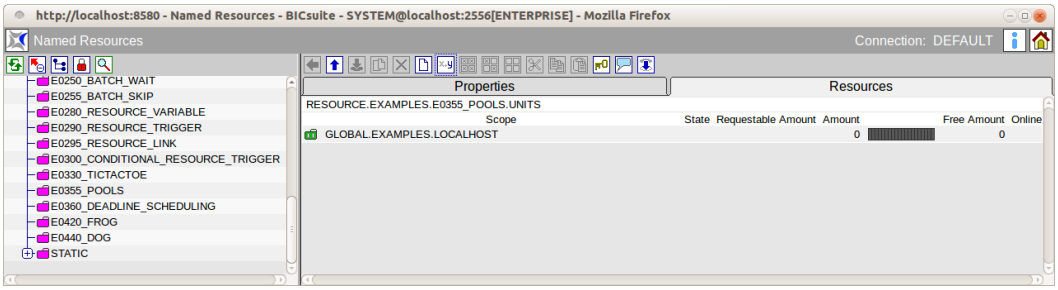


Abbildung 10.10: Named Resources; Pool Instanzen

10.4 Selektor Named Resource

Dieser Navigator dient zur Ein- oder Mehrfachauswahl von Named Resource Definitionen und wird von anderen Dialogen z. B. Footprint als Such- und Selektionsmaske aufgerufen. Die angezeigten Ressourcen sind abhängig vom aufrufenden Dialog. Wird der Selektor aus dem Footprint Editor aufgerufen, werden nur System Resources angezeigt. Wird der Selektor aus dem Environment aufgerufen, werden nur statische Ressourcen angezeigt.

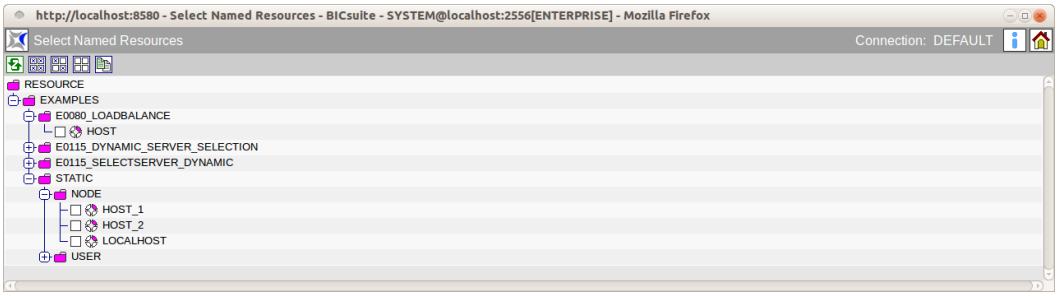


Abbildung 10.11: Named Resources; Selektor

11 Environments

11.1 Bild

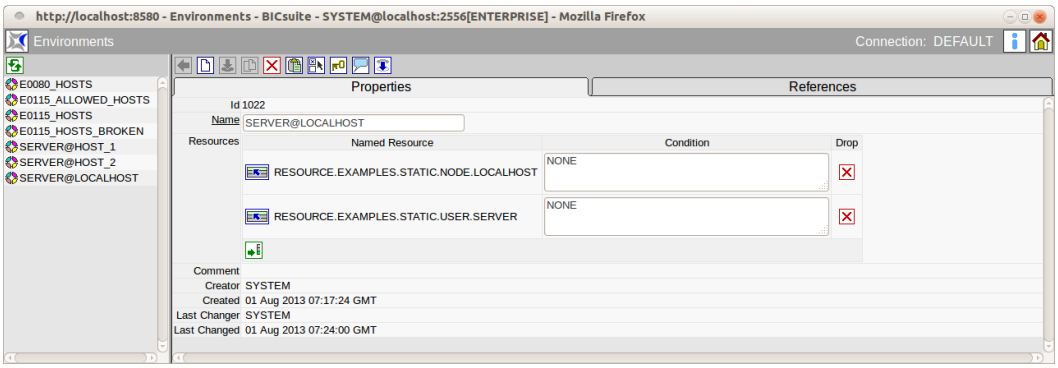


Abbildung 11.1: Environments

11.2 Konzept

11.2.1 Kurzbeschreibung

Der Dialog ”Environment” dient zum Verwalten von Umgebungsdefinitionen. Eine Environment fasst mehrere Anforderungen statischer Ressourcen unter einem Namen zusammen.

11.2.2 Ausführliche Beschreibung

Die Environments dienen zwei Zwecke. Einerseits dienen sie zur vereinfachten Pflege von Resource-Anforderungen, da sie nicht mehr einzeln innerhalb der Job Definitions hinzugefügt werden müssen. Es kann dafür das jeweilige Environment angegeben werden. Andererseits wird mittels Environments festgelegt, welche Benutzer bzw. welche Benutzergruppe bestimmte Ablaufumgebungen benutzen dürfen. Die Environment Definition gibt also an, welche Laufzeitumgebung ein Job benötigt. Zum Beispiel kann über ein Environment gesteuert werden, auf welchem Host ein Job laufen soll oder welche Benutzer und Programme vorhanden sein müssen.
Das Environment ist ein Pflichtparameter in einer Job Definition.

Environments können auch als Folder Environment bei einem Folder eingetragen werden. Alle unter einem solchen Folder liegenden Job Definitionen ”erben” dadurch die Resource Anforderungen des Folder Environments.
Gibt es zum Beispiel jeweils einen Job Server für Entwicklung und Produktion, kann am Job ein Environment verwendet werden, welches beide Job Server als Ausführungsort zulässt. Werden nun für Entwicklung und Test zwei Folder angelegt, welchen jeweils ein Environment ”Entwicklung” bzw. ”Produktion” zugeordnet wird, deren Resource Anforderungen nur von Entwicklungs- bzw. Produktions Jobs Servern erfüllt werden kann, wird damit eine leicht bedienbare Trennung von Entwicklungs- und Produktionsumgebung erreicht. Befindet sich ein solcher Job irgendwo unter dem Folder ”Entwicklung”, führt die Kombination der Resource Anforderungen von Job Environment und Folder Environment zur Auswahl des richtigen Entwicklungs Job Servers. Wird der Job unter den Folder ”Produktion” verschoben oder kopiert und dann ausgeführt, so wird automatisch der Produktions Job Server ausgewählt. Eine Änderung der Job Definition ist nicht notwendig.

11.3 Navigator



Abbildung 11.2: Environments Navigator

Der Navigationsbildschirm des ”Environment” Dialoges zeigt vorhandenen Environments an.
Ist der angemeldete Benutzer Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage environment” Privileg, so werden alle Environments angezeigt. Ist dies nicht der Fall, so werden nur Environments angezeigt, die vom aktuell angemeldeten Benutzer genutzt werden dürfen. Das bedeutet, der angemeldete Benutzer muss Mitglied einer Gruppe sein welche das ”use” Privileg auf das jeweilige Environment besitzt.

11.4 Editor

11.4.1 Tab Properties

Dieser Tab dient der Pflege der Environmenteigenschaften. Environments dürfen nur von Benutzern, welche Mitglied der Gruppe ”ADMIN” oder einer anderen

Gruppe mit dem ”manage environment” Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder ”read only”.
Er sieht folgendermaßen aus:

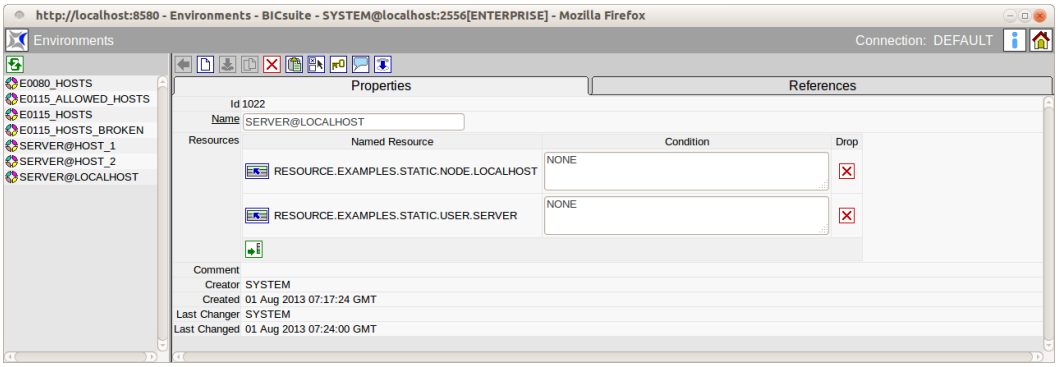


Abbildung 11.3: Environment Properties

Der Properties-Tab für Environments hat folgende Felder:

ID Die ID wird automatisch vergeben und dient zur eindeutigen systemweiten Identifikation des Objektes.

Name Der Name des Environments. Dieser ist frei wählbar und muss innerhalb des Environments eindeutig sein.
Die Liste ”Resources” zeigt alle zum Environment gehörenden Ressourcen an. Es handelt sich hier ausschließlich um statische Resources. Das Feld *Named Resource* hat folgende Bedeutung:

Named Resource Hier wird der Name der Named Resource angezeigt.

Condition Im Feld *Condition* wird eine Bedingung eingetragen, die erfüllt sein muss, damit die Resource als gültig erkannt wird. Die Condition wird im Kontext des anfordernden Jobs ausgewertet.

Drop Mit dem *Drop*-Button kann die betreffende Zeile gelöscht werden.

11.4.2 Tab References

Dieser Tab dient der Anzeige welche Referenzen auf das Environment bestehen. Er sieht folgendermaßen aus:
Der References Tab zeigt eine Baumansicht welche nur die Folder enthält unter welchen sich Objekte befinden, die das Environment referenzieren.



Abbildung 11.4: Environment References

Folder werden normalerweise kursiv dargestellt, es sei denn, der Folder referiert das Environment als Folder Environment.
Ein Mausklick auf den Namen eines nicht rekursiv dargestellten Folders oder eines Jobs, öffnet ein Editorfenster für den Folder bzw. Job.

12 Footprints

12.1 Bild

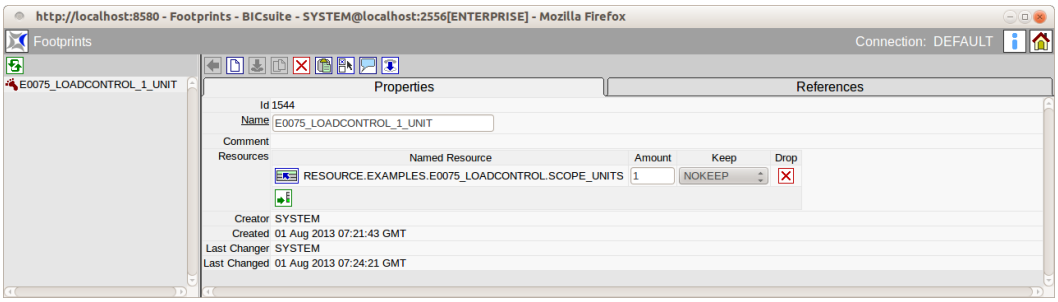


Abbildung 12.1: Footprints

12.2 Konzept

12.2.1 Kurzbeschreibung

Dieser Dialog dient zum Erstellen einer Footprint Definition. Ein Footprint fasst eine Gruppe Anforderungen von System Resources unter einem Namen zusammen.

12.2.2 Ausführliche Beschreibung

Die Zusammenfassung von Resource-Anforderungen zu Footprints, dient der leichteren Pflege der Anforderungen von Jobs an die Laufzeitumgebung, da sie nicht einzeln spezifiziert werden müssen.
Der Footprint kann für jeden Job übersteuert werden. Die Übersteuerung ist durch eine explizite Anforderung einer (im Footprint enthaltenen) Resource mit einer abweichenden Menge oder unterschiedlicher Einstellung des Keep Parameters möglich.
Eine Resource-Anforderung kann nicht entfernt werden, d.h. alle im Footprint definierten Ressourcen werden angefordert. Die Menge kann bis zum Wert 0 vermindert werden, allerdings wird trotzdem das Vorhandensein der Resource gefordert.
Beispiel:
Im Footprint wird eine System Resource CPU_UNIT mit der Menge 4 (CPU Einheiten) definiert. Durch Hinzufügen einer Anforderung der selben System Resource

(CPU_UNIT) innerhalb der Job Definition (Tab Resources) mit einer Menge von 2 (CPU Einheiten), kann der Footprint übersteuert werden. Es ist aber nicht möglich die Resource CPU_UNIT ganz zu entfernen.

12.3 Editor

12.3.1 Tab Properties

Dieser Tab dient der Pflege der Footprinteigenschaften.
Footprints dürfen nur von Benutzern, welche Mitglied der Gruppe "ADMIN" oder einer anderen Gruppe mit dem "manage footprint" Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder "read only".
Er sieht folgendermaßen aus:

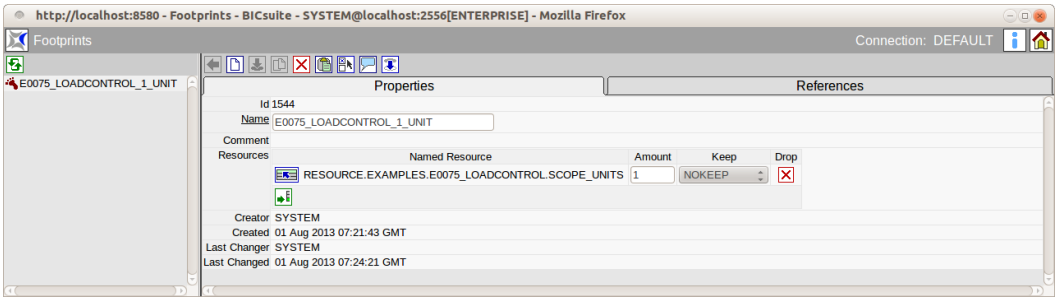


Abbildung 12.2: Footprint Properties

Der Properties-Tab für Footprints hat folgende Felder:

ID Die ID wird automatisch vergeben und dient der eindeutigen systemweiten Identifikation des Footprints.

Name Der Name des Footprints. Dieser ist frei wählbar.
In der Liste "Resources" kann eine Liste für die enthaltenen Ressourcen des Footprints angegeben werden.
Die Felder haben folgende Bedeutung:

Named Resource Hier wird der Name der Named Resource angezeigt. Die Auswahl erfolgt über einen *Chooser* Button.

Amount Das ist die Menge der Resource die ein Job benötigt.

Keep Mittels des Keep Parameters wird bestimmt ob die Named Resource nach Ablauf eines Jobs gehalten wird oder wieder freigegeben werden kann. Es gibt folgende Auswahlmöglichkeiten:

1. *No_Keep*

   Die Named Resource wird nach Beendigung des Jobs freigegeben. Ob der Job erfolgreich beendet oder wegen eines Fehlers abgebrochen wurde, spielt für die Freigabe keine Rolle.

2. *Keep*

   Die Resource wird erst freigegeben, wenn der Job einen finalen Status erreicht hat. Im Fehlerfall (Restartable State) wird die Resource gehalten.

3. *Keep_Final*

   Die Resource wird erst freigegeben, wenn der Job und alle seine Children einen finalen Status erreicht haben.

12.3.2 Tab References

Dieser Tab dient der Anzeige welche Referenzen auf den Footprint bestehen. Er sieht folgendermaßen aus:

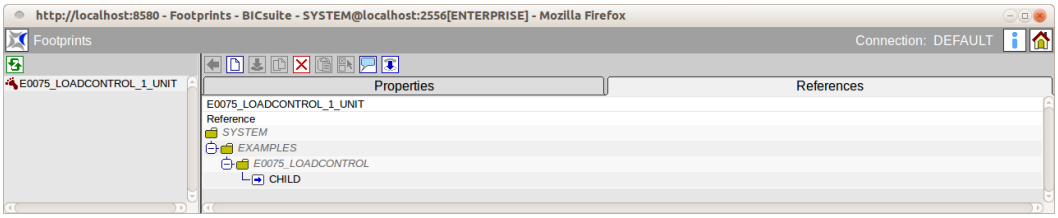


Abbildung 12.3: Footprint References

Der References Tab zeigt eine Baumansicht welche nur die Folder enthält unter welchen sich Jobs befinden, welche den Footprint referenzieren.
Folder werden kursiv dargestellt.
Ein Mausklick auf den Namen eines Jobs, öffnet ein Editorfenster für den Job.

13 Jobservers and Resources

13.1 Bild

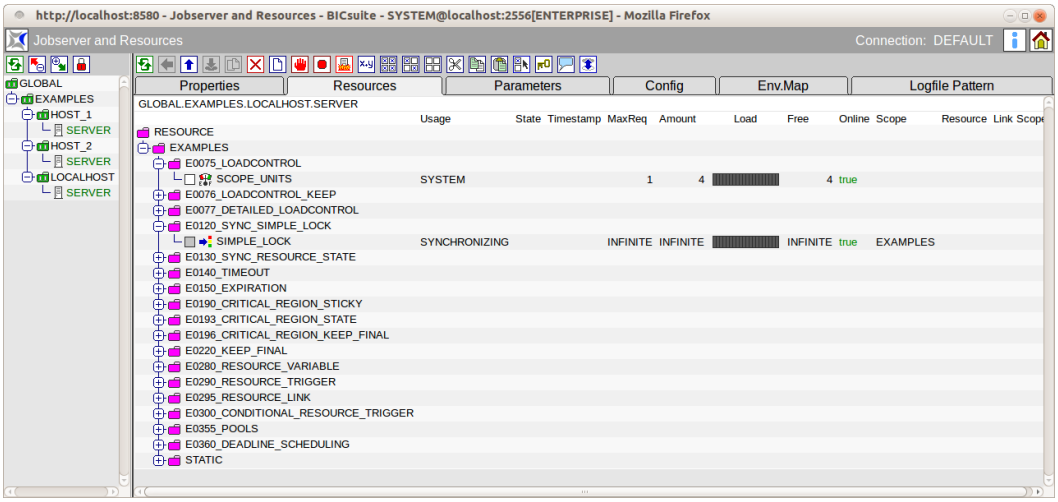


Abbildung 13.1: Jobservers and Resources

13.2 Konzept

13.2.1 Kurzbeschreibung

Ein Jobserver ist ein BICsuite Systemprozess, der auf allen Rechnern die als ausführende Einheiten des BICsuite Scheduling Systems definiert wurden, laufen muss. Wird einem Jobserver durch den BICsuite Server ein Ablaufobjekt zum Ausführen zugeteilt, führt der Jobserver das in der Job Definition angegebene Skript oder Programm in der notwendigen Umgebung mit den notwendigen Parametern aus. Der Jobserver überwacht das Programm und gibt den Rückkehrstatus des Skripts oder Programms an den BICsuite Server zurück.

13.2.2 Ausführliche Beschreibung

Im Dialog "Jobserver and Resources" kann die physikalische Aufteilung definiert und gepflegt werden, d. h. welche Jobserver auf welchen Maschinen laufen und

welche Ressourcen von diesen Jobservern zur Verfügung gestellt werden.
Die Einrichtung und der Start dieser Jobserver muss auf Systemebene durchgeführt werden.
Im Dialog werden zwei Arten von Objekten unterschieden: einerseits gibt es so genannte Scopes, die als Container für die jeweiligen Jobserver oder untergeordnete Scopes dienen können.
Andererseits werden in diesem Dialog die Jobserver verwaltet, die die ausführenden Systemprozesse widerspiegeln.
Die Unterteilung der Hierarchie ist frei durchführbar und hängt größtenteils von der Komplexität der unterlagernden Systemumgebung und Verwaltungsstruktur (z. B. im Rechenzentrum) ab.
Beispielsweise kann jeder physikalische Hostrechner als Scope und alle auf diesem Rechner, benötigte Ablaufumgebung als Jobserver unterhalb dieses Scopes definiert werden. Falls die Systemlandschaft komplexer ist, sind weitere Ebenen für Abteilungen, Systemarchitekturen (Unix, Windows) oder ähnliches hilfreich.
Auf jeder Hierarchieebene können den Scopes und Jobservern Ressourcen zugeordnet werden. Diese Ressourcen stehen dann allen untergeordneten Jobservern zur Verfügung.
Beispiel:
Auf der Ebene eines Hosts wird die System Resource CPU_UNITS mit einem Wert von 5 eingetragen. Werden auf dem Rechner 2 Jobserver angelegt, stehen diesen Jobservern diese 5 CPU Einheiten gemeinsam zur Verfügung. Belegt ein Jobserver durch die Ausführung eines Jobs 3 CPU Einheiten, stehen dem anderen Jobserver nur noch 2 CPU Einheiten zur Verfügung. Dieses Verhalten ist jedoch nicht additiv. Wird auf einem der obengenannten Jobserver ebenfalls die Resource CPU_UNITS angelegt, etwa mit einem Wert von 3, stehen diesem Jobserver ausschließlich diese 3 CPU_UNITS zur Verfügung. Dem anderen Jobserver stehen demnach dann 5 CPU_UNITS vom übergeordneten Scope zur Verfügung.

13.3 Navigator

Der Navigator bietet eine hierarchische Darstellung, in der Scopes als Ordner definiert werden. Innerhalb der Scopes befinden sich die Jobserver.

13.4 Editor

13.4.1 Tab Properties

Der Tab Properties dient zur Pflege der Eigenschaften von Scopes und Jobservern.
Tab ”Properties” für Scopes ist dargestellt in Bild 13.3.
Tab ”Properties” für Jobserver ist dargestellt in Bild 13.4.
Die Felder des Tabs ”Properties” haben folgende Bedeutung:

Editor

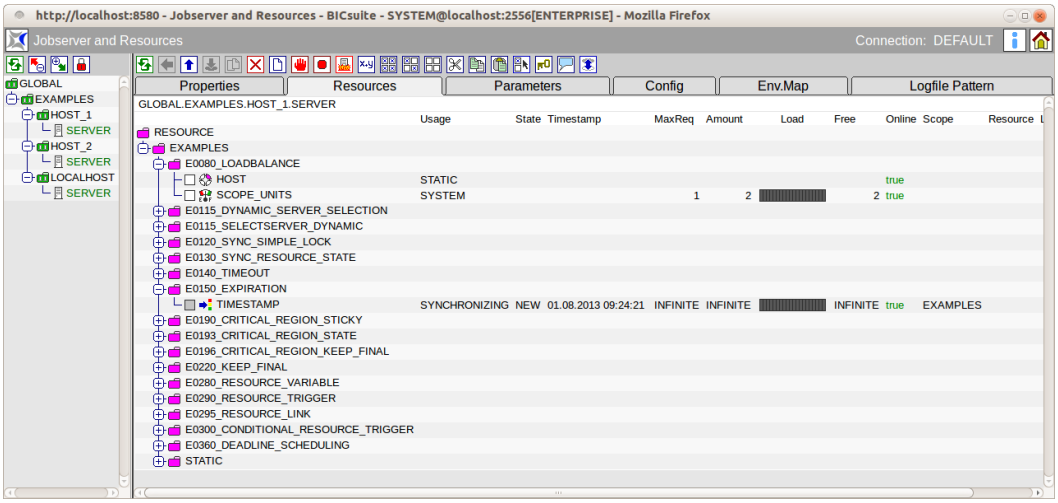


Abbildung 13.2: Jobservers und Scopes Navigator

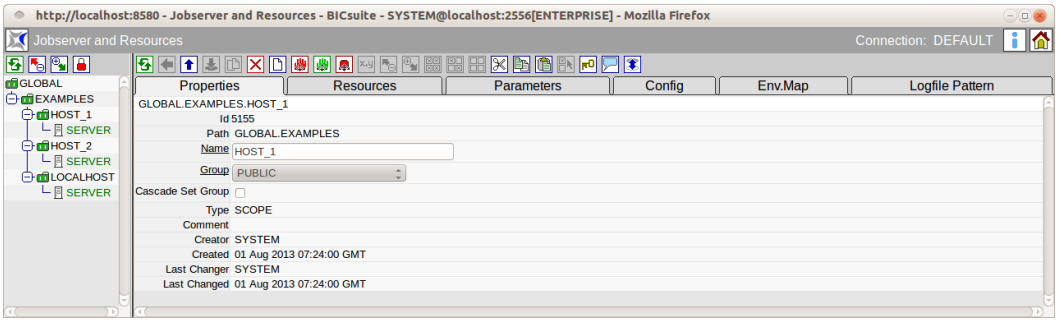


Abbildung 13.3: Scope Properties

ID Die ID dient zur eindeutigen systemweiten Identifikation des Objektes.

Path (Pfad) Der Path stellt den kompletten Pfad des Objektes innerhalb der Hierarchie dar.

Name Das ist der Name des Scopes oder des Jobservers. Der Name kann frei gewählt werden, muss aber innerhalb der Hierarchiestufe eindeutig sein.

Group Die Gruppe kann aus einer ”Drop Down” Liste ausgewählt werden. Sie bezeichnet die Owner-Gruppe des Scopes.

Type Beim Feld *Type* handelt es sich um den Typ des selektierten Objektes. Der Typ wird bei der Neuanlage eines Servers oder Scopes gewählt und kann dann

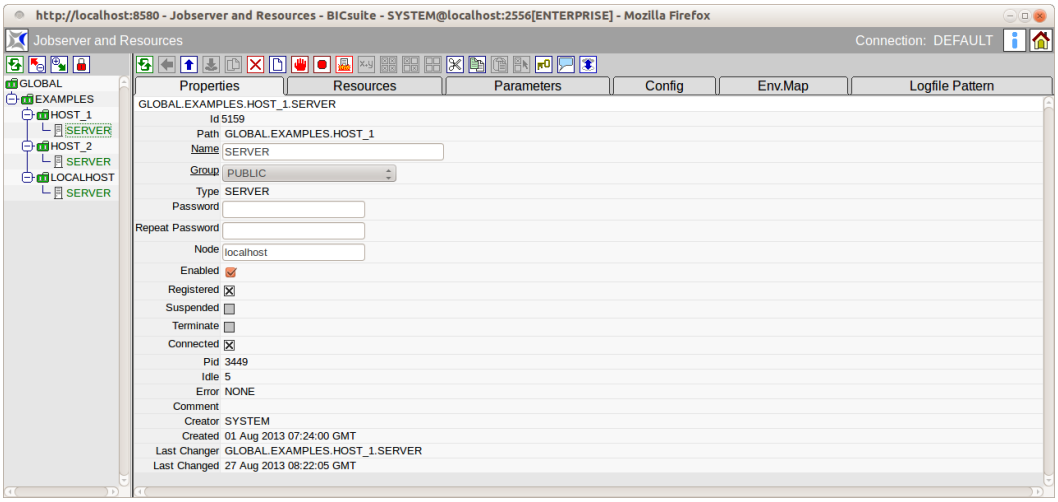


Abbildung 13.4: Jobserver Properties

nicht mehr verändert werden.
OPTIONEN FÜR TYPE

1. *Scope*

   Ein Scope ist eine Kategorie unter der Jobserver gruppiert und hierarchisch geordnet werden können. Dies könnte zum Beispiel ein Hostrechner sein oder auf noch höherer Ebene, alle Rechner der Abteilung oder alle Windows-basierten Rechner. Der Aufbau und die Tiefe der Hierarchie kann je nach Systemlandschaft angepasst werden und sollte diese widerspiegeln.

2. *Server*

   Bei dem Objekt handelt es sich um einen physikalischen Jobserver. Der physikalische Prozess muss sich unter dem gewählten Namen beim BICsuite System anmelden und damit die Verbindung zu dieser Jobserverdefinition schaffen.

Password Das *Password* dient zum Anmelden des Jobservers am BICsuite Server. Jeder Jobserver benötigt seinen Namen und das Passwort um sich am BICsuite Server anmelden zu können. Nur mit gültiger Anmeldung kann ein Job an den Jobserver zur Ausführung vergeben werden.
Das Passwort wird in der Anzeige verdeckt und muss mit dem Feld *Repeat Password* identisch sein.
Da das Passwort im Dialog nicht angezeigt wird, ist es nicht ersichtlich ob ein Passwort bereits eingegeben wurde oder dies noch notwendig ist. Bei Änderung von anderen Eigenschaften muss das Passwort nicht erneut spezifiziert werden.

Falls der Jobserver angemeldet ist, wird er automatisch von eventuellen Änderungen eines Passwortes in Kenntnis gesetzt.

Repeat Password Das *Repeat Password* muss bei der Eingabe des Passwortes identisch zum Feld *Password* sein. Die doppelte Eingabe ist nötig, um versehentliche Tippfehler festzustellen.

Node Der *Node* gibt an, auf welchem Rechner der Jobserver läuft. Dieses Feld hat einen rein dokumentativen Charakter.

Enabled Falls der Jobserver das Enabled Flag gesetzt hat, ist eine Anmeldung des Jobserverprozesses am BICsuite Server möglich. Ist das Flag nicht gesetzt, schlägt die Anmeldung fehl.

Suspended Ist das Suspended Flag gesetzt, kann sich der Jobserverprozess zwar anmelden, es werden ihm aber keine Jobs zur Ausführung zugeteilt. Statusänderungen von bereits zugeteilten Jobs werden vom BICsuite Server entgegengenommen.

Terminate Das Terminate Flag zeigt an, ob der Jobserver sich terminieren soll. Wenn das Flag gesetzt ist, wird der Jobserver diese Nachricht beim nächsten Kommunikationsschritt erhalten und sich entsprechend verhalten.
Die Terminierung des Jobservers kann einige Sekunden dauern. Die Beendigung ist mittels des *Refresh* Buttons überprüfbar. Wurde der Jobserver beendet, so wechselt die Anzeige des Feldes *Connected* auf den Wert "FALSE".

Connected Das Feld *Connected* gibt Auskunft über den Verbindungsstatus eines externen Jobserver-Prozesses. Wurde die Anmeldung des Jobserverprozesses am BICsuite System erfolgreich abgeschlossen, steht dieser Wert auf "TRUE".

PID Bei der PID handelt es sich um die Prozess-Identifikationsnummer des Jobserverprozesses auf dem jeweiligen Hostsystem. Sie wird während der Registrierung dem BICsuite System mitgeteilt. Über die PID und das *Node*-Feld ist es möglich, den Prozess auf dem Betriebssystem des jeweiligen Servers zu lokalisieren.

Error Ist bei der Ausführung des Jobserverprozesses ein Fehler aufgetreten, wird hier eine Fehlermeldung angezeigt.

13.4.2 Tab Resources

Der Tab Resources sieht folgendermaßen aus:
Im Tab Resources werden alle Ressourcen angezeigt, die der aktuelle Scope oder Jobserver anbietet. Sollen Ressourcen, insbesondere vom Typ Synchronizing, systemweit vergeben werden, kann dies im Scope "GLOBAL" geschehen.

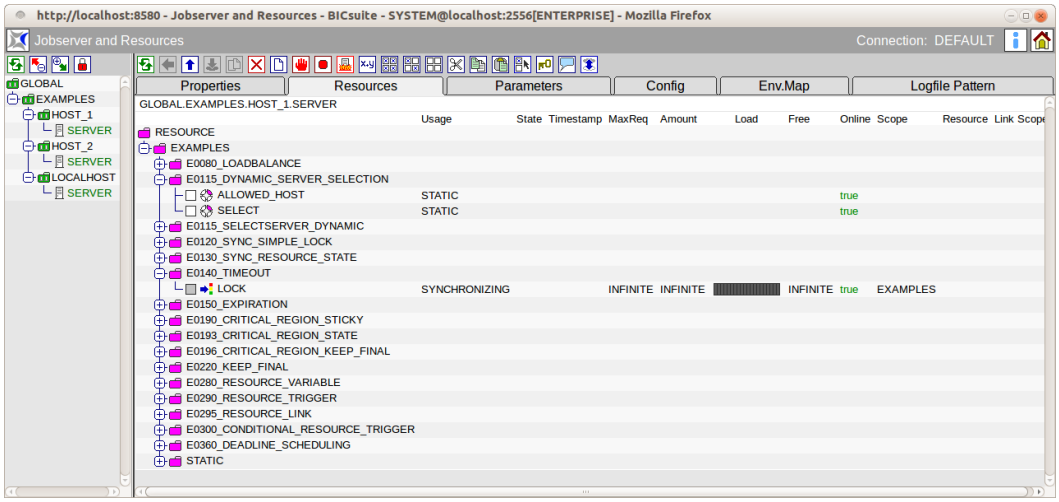


Abbildung 13.5: Jobserver und Scope Ressourcen

Werden Ressourcen mehrmals, innerhalb von unterschiedlichen Scopes oder Jobservern, eingetragen, handelt es sich hierbei um eine jeweils eigene Instanz der Resource.
Beispiel:
Wird eine Synchronizing Resource A in Scope X und Scope Y eingetragen, sind beide Instanzen der Resource A (in X und Y) voneinander unabhängig. Das heißt, eine mögliche Synchronisation findet nur innerhalb aller Jobs in Scope X und innerhalb aller Jobs in Scope Y statt. Soll hier eine globale Synchronisation stattfinden, muss die Resource A im Scope GLOBAL definiert werden.
Die Anzeige der Ressourcen findet als Abbildung der Hierarchie aus der Navigation des Dialoges Named Resources statt, wobei zusätzliche Informationen angezeigt werden.
Das erste Feld der Liste gibt den Namen der Named Resource oder der Kategorie an. Handelt es sich um eine Kategorie, erscheint zusätzlich noch ein Ordner-Icon. Durch Anklicken dieses Icons kann die Kategorie geöffnet und geschlossen werden. Handelt es sich um eine Named Resource, kann der Name angeklickt werden und führt zum Tab Resource Detail.
Die restlichen Felder werden werden im nächsten Abschnitt erläutert.

13.4.2.1 Tab Resource Detail

Der Tab Resource Details wird angezeigt, sobald eine Resource ausgewählt wurde. Der Tab dient zur Eingabe und Pflege der Instanzinformationen für diese Named Resource. Mit Instanz ist hier die Ausprägung der Named Resource im gewählten Scope bzw. Jobserver gemeint. Alle Instanzparameter können hier eingegeben und gepflegt werden.

Der Tab Resource Detail sieht folgendermaßen aus:

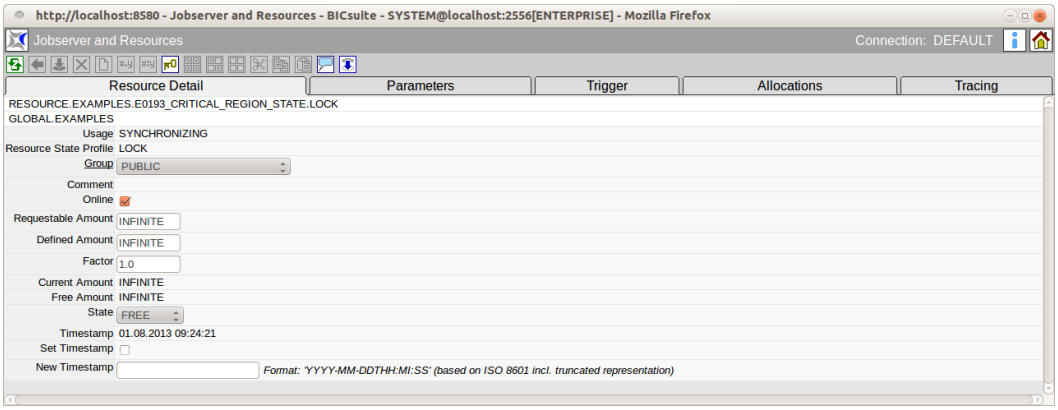


Abbildung 13.6: Resource Details

Die Felder des Tabs ”Resource Detail” haben folgende Bedeutung:

Usage Die *Usage* gibt an, um welchen Typ ”Resource” es sich handelt. Mehr zu Typen von Ressourcen finden Sie im Dialog Named Resource.

Resource State Profile Dieses Feld ist nur zu sehen, wenn die Usage der Resource ”Synchronizing” ist.
Es handelt sich hier um das zur Resource zugeordnete Resource State Profile.

Online Mittels *Online* ist es möglich eine Resource in diesem Scope oder Jobserver on- bzw. offline zu schalten. Ist die Resource offline, so steht diese Resource vorübergehend nicht zur Verfügung.
Beispiel:
Eine geplante Downtime eines Rechners oder einer Datenbank steht an. Wurde der Rechner oder die Datenbank als Resource im System abgebildet, kann die Resource, sobald sie physikalisch nicht mehr verfügbar ist, offline gesetzt werden. Alle Jobs die diese Resource benötigen, werden automatisch nicht mehr ausgeführt. Sollte die Resource in einem anderen Scope ebenfalls definiert sein (als eigene Instanz), so würden alle Jobs die die Resource benötigen auf diesen Scope oder Jobserver ausweichen.
Ist eine Resource offline, führt dies *nicht* zu einer ”Job cannot run in any scope because of resource shortage” Fehlermeldung.

Requestable Amount Der *Requestable Amount* ist die Menge die maximal angefordert werden kann.

Defined Amount Dieses Feld ist nur zu sehen, wenn die Usage der Resource "Synchronizing" oder "System" ist.
Der *Defined Amount* gibt die aktuelle Anzahl von Instanzen der Named Resource für diesen Scope oder Jobserver an. Sollten sich Änderungen in der Anzahl für diesen Scope ergeben, so können sie mit diesem Feld angepasst werden.
Beispiel:
Dies kann bei Hardware-Änderungen unter Umständen notwendig werden. Stellt die Resource eine CPU Einheit dar und sind im aktuellen Host 2 CPU Einheiten verbaut, so ist der Amount 2. Werden nun 2 weitere CPU Einheiten eingebaut, muss der Amount-Wert auf 4 erhöht werden.

Current Amount Der *Current Amount* und der *Defined Amount* sind für nicht pooled Resource identisch.

Free Amount Dieses Feld ist nur zu sehen, wenn die Usage der Resource "Synchronizing" oder "System" ist.
Der *Free Amount* bezeichnet die Anzahl aller noch nicht von Jobs belegten Instanzen einer Resource, innerhalb des gewählten Scopes oder Jobservers.

Status/State Das Feld *State* ist nur zu sehen, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde.
Hierbei handelt es sich um den aktuellen Status der Resource in diesem Scope oder Jobserver. Der Status kann mittels dieses Wertes gesetzt bzw. geändert werden. In der "Drop Down" Liste sind alle gültigen Resource States des Resource State Profiles enthalten und können ausgewählt werden.
Dies kann bei einer manuellen Fehlerbehebung notwendig sein, um eine Resource wieder in einen Status zu bringen, in dem eine Weiterverarbeitung durch Folgejobs möglich ist.
Falls hier der Wert manuell geändert wird, wird der Wert des Feldes *State* im Tab Resources nicht automatisch aktualisiert. Falls dies gewünscht und notwendig ist, kann dies mittels des *Refresh* Buttons durchgeführt werden.

Timestamp Dieses Feld ist nur zu sehen, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde.
Der *Timestamp* gibt die Zeit des letzten Statuswechsels einer Resource an.
Timestamps spielen eine Rolle, falls in einem Ablaufobjekt ein Expiration-Intervall angegeben wurde. Ist dies geschehen, darf der Timestamp nicht älter sein als das angegebene Intervall. Mehr zu Expiration-Zeiten finden Sie im Kapitel 14.5.8.1.
Wird durch die Beendigung eines Jobs eine Änderung des Resource State hervorgerufen, wird der Timestamp automatisch vom BICsuite Server aktualisiert.

Set Timestamp Dieses Feld ist nur zu sehen, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde. Der *Set Timestamp* Flag ist ein sog. Action-Flag. Wird das Flag gesetzt und anschließend der *Speichern*-Button gedrückt, so wird die Zeit des *Timestamp*-Feldes auf die aktuelle Zeit gesetzt. Beispiel:
Dies kann notwendig werden, wenn ein manueller Eingriff bei einer Fehlerbehebung notwendig geworden ist und dabei die Statusänderung manuell über das Feld *State* durchgeführt wurde.

Factor Um Resource Anforderungen von Jobs von außen justieren zu können, wurde ein Resource Factor eingeführt. Dieser kann sowohl an der Named Resource als auch individuell an der Resource gesetzt werden. Bei der Bestimmung ob ein Job eine bestimmte Resource belegen kann wird durch den Vergleich der ursprünglichen Anforderung mit dem Requestable Amount bestimmt. Bei der tatsächlichen Belegung wird jedoch

ceil(Anforderung * Factor)

hergenommen. Der Default Wert ist natürlich 1.

13.4.2.2 Tab Parameters

Im Tab Parameters können zusätzliche Informationen, die von Jobs oder Resource Trigger ausgewertet werden können, zu einer Resource gespeichert werden. Die Definition der Parameters erfolgt bei den Named Resources.

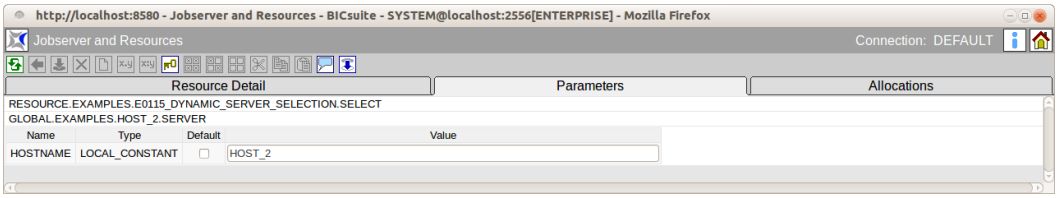


Abbildung 13.7: Resource Parameters

Die Spalten der obigen Liste haben folgende Bedeutung:

Name In der Spalte "Name" wird der Name des Parameters angezeigt.

Type Beim Type handelt es sich um die Art des Parameters. Es gibt folgende Möglichkeiten:

- Constant: Der Constant hat einen festen Wert und gilt für alle Resources.
- Local Constant: Der Local Constant hat einen festen Wert, der von Resource zu Resource abweichen kann.

Die Werte der Local Constants sind an dieser Stelle änderbar.

Default Das Feld *Default* gibt an ob der angezeigte Wert der Default-Wert ist. Soll der Wert auf dem Default-Wert zurückgesetzt werden müssen, muss das Flag gesetzt werden.

Value Der Value ist der Wert des Parameters.

13.4.2.3 Tab Allocations

Der Tab Allocations gibt Informationen, welche Tasks aktuell die Resource belegen und welche Tasks Resources benötigen, sie aber etwa aufgrund eines falschen Status oder wegen eines Locks nicht bekommen können.
Der Tab Sheet sieht folgendermaßen aus:

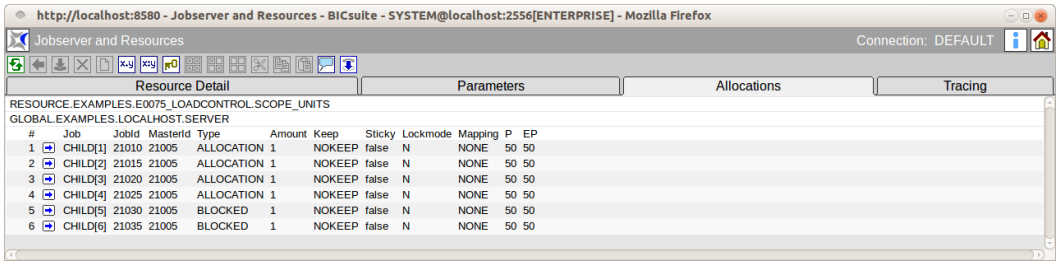


Abbildung 13.8: Resource Allocations

Die Spalten der obigen Liste haben folgende Bedeutung:

Job Je nach Einstellung wird hier entweder der Name des Jobs oder der gesamte Pfad der Ordnerhierarchie für diesen Job angezeigt.

Job Id Hierbei handelt es sich um die ID der Jobinstanz, welche durch ein direktes Submit des Jobs im Dialog Submit Jobs oder durch ein Submit des Master Batches oder Jobs gestartet wurde.

Master Id Hierbei handelt es sich um die ID der Job- oder Batchinstanz, welche als Master Job im Dialog Submit Jobs gestartet wurde und den aktuellen Job als Child beinhaltet. Handelt es sich bei dem Job selber um einen Master Job, wird die eigene Job Id angezeigt.

Type Beim *Type* handelt es sich um die Art wie die aktuelle Jobinstanz auf die Resource zugreift bzw. zugreifen will. Es gibt folgende Optionen:

1. Requested

   Der Job fragt beim Server an ob die Resource verfügbar ist.

2. Reserved

   Die Resource ist vom Job reserviert. Dies hat die gleiche Auswirkung wie Allocated, nur können reservierte Ressourcen beim Job Start wieder freigegeben werden. Belegte (allocated) Ressourcen werden frühestens beim Job End wieder freigegeben.

3. Allocated

   Der Typ "Allocated" gibt an dass der Job aktuell läuft und die Resource im Zugriff hat. Andere Jobs, welche auf diese Resource zugreifen wollen, können abhängig vom Typ der Zugriffssperre diese ebenfalls belegen oder werden blockiert.

4. Blocked

   Ist ein Job mit dem Typ "Blocked" gekennzeichnet, kann dieser im Moment nicht ausgeführt werden, da er auf den Zugriff der gewählten Resource wartet, diese aber wegen eines nicht passenden Lockmode oder anderer Kriterien nicht bekommen kann.

   Der Grund weshalb die Resource nicht zur Verfügung steht wird durch rote Schrift gekennzeichnet.

   Sobald die Resource wieder verfügbar ist, wechselt der Typ von Blocked in Requested.

5. Available

   Die Resource steht zur Verfügung.

6. Ignored

   Der Anwender hat das System beauftragt die Resource-Anforderung zu ignorieren.

7. Master Reservation

   Eine Master Reservation ist eine Reservierung von Sticky Resources.

Amount Beim *Amount* handelt es sich um die Anzahl der vom aktuellen Job belegten oder angeforderten Resource-Instanzen. Übersteigt der Wert des Amounts die aktuell verfügbare Anzahl die im Feld *Amount* im Tab Resource Detail gepflegt werden kann und ist kein alternativer Scope mit einer ausreichenden Anzahl vorhanden, kann der Job nicht ausgeführt werden.

Keep Der *Keep*-Parameter gibt an, mit welchem Keep-Status die aktuelle Resource gehalten wird. Weitere Informationen zum Keep-Status finden Sie im Kapitel 12.3.1.

Sticky Der *Sticky*-Parameter gibt an, ob der Job die aktuelle Resource "Sticky" hält oder nicht. Weitere Informationen zum Sticky Flag finden Sie im Kapitel 14.5.8.1.

Lockmode Der *Lockmode* gibt an mit welchem Zugriffsmodus die Resource vom aktuellen Job belegt wird. Es gibt folgende Zugriffsmodus:

1. Exclusive (X)

   Kein anderer Job hat Zugriff auf diese Resource.

   Beispiel: Laden einer Datenbanktabelle.

2. Shared Exclusive (SX)

   Diese Zugriffe sind untereinander verträglich, jedoch unverträglich mit Shared-Zugriffe.

   Beispiel: Eine Anwendung führt viele kleine schreibende Transaktionen auf einer Datenbanktabelle durch.

3. Shared (S)

   Shared-Zugriffe sind untereinander verträglich.

   Beispiel: Auswertungen über komplette oder große Teile von Datenbanktabellen.

4. Shared Compatible (SC)

   Diese Zugriffe sind sowohl untereinander als auch mit S- und SX-Zugriffe verträglich.

   Beispiel: Eine Anwendung führt viele kurzlaufende, lesende Transaktionen auf einer Datenbanktabelle durch. Solche Anwendungen behindern sich nicht gegenseitig. Auch kleine schreibende Transaktionen, sowie große Auswertungen stellen keine Behinderung dar.

5. NoLock (N)

   Der Lockmode NoLock gibt an, dass weder eine Sperrung der Resource durchgeführt wird, noch beachtet werden soll. Ein Job der eine Resource mit NoLock sperrt, kann immer auf diese zugreifen, unabhängig von anderen Sperroptionen.

Hier noch einmal zusammenfassend die Kompatibilitätsmatrix aller Sperroptionen untereinander.

(+) bedeutet die Lockmodes sind kompatibel zueinander. Zwei Jobs, welche die beiden Lockmodes anfordern, können zur selben Zeit laufen.
(-) bedeutet die Lockmodes schließen sich aus. Ein Job wird blockiert, bis der andere Job die Resource wieder freigibt.

	Exclusive (X)	Shared (S)	Shared Exclusive (SX)	Shared Compatible (SC)	NoLock (N)
Exclusive	-	-	-	-	+
Shared	-	+	-	+	+
Shared Exclusive	-	-	+	+	+
Shared Compatible	-	+	+	+	+
NoLock	+	+	+	+	+

Tabelle 13.1: Lockmode-Matrix

Mapping Das *Mapping* gibt das aktuelle Resource State Mapping an, welches die Resource verwendet.

P (Priorität) Die *Priorität* gibt an, welche Startpriorität der Prozess beim Submit im Dialog Submit Jobs hatte. Der Wertebereich der Priorität beginnt bei 100 (niedrigste Priorität) und endet bei 0 (höchste Priorität)

EP (Effektive Priorität) Je länger ein Job auf die Ausführung im System wartet, desto höher wird die effektive Priorität der Ausführung im BICsuite System. Früher gestartete Jobs sollen damit vor den später gestarteten Jobs berücksichtigt werden (je nach Einstellungen im Feld Priority im Batches-and-Jobs-Dialog und der Startzeit).
In der Serverkonfiguration ist einstellbar, wie die Priorität über die Zeit steigt. Standardmäßig ist eingestellt, dass mit jeder halben Stunde, die seit dem Submit des Jobs vergangen ist, die Priorität um einen Punkt steigt.
Als Beispiel: Wurde ein Job um 12 Uhr mit der Priorität 50 gestartet, so ist die effektive Priorität um 12:30 Uhr 49, um 13:00 Uhr 48 usw. bis die höchste Stufe erreicht wurde.

Blockierte Prozesse Ist der aktuelle Job in der Liste als blockiert markiert, erscheint der Grund warum der Job nicht ausgeführt werden kann in rot.
Beispiel: Wird die Resource gerade exklusiv von einem anderen Job belegt und der aktuelle Job möchte die Resource ebenfalls exklusiv belegen, ist das Feld *Lockmode* rot markiert.

13.4.2.4 Tab Tracing

Im *Tab Tracing* kann man die Resourcebelegung zurückverfolgen oder sie überwachen.

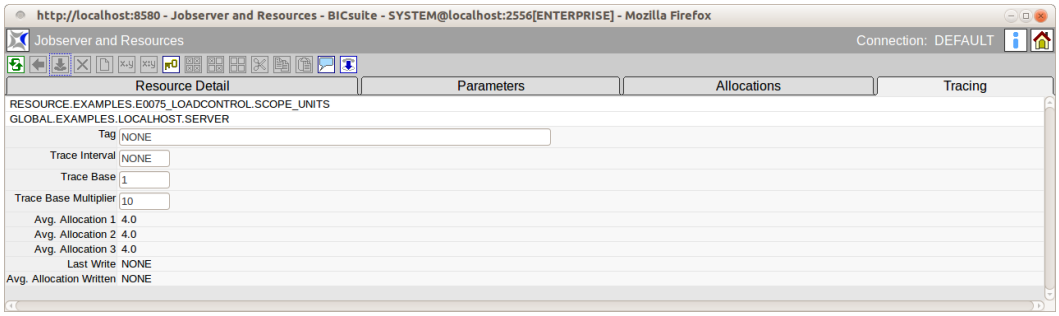


Abbildung 13.9: Resource Tracing

Tag Name unter dem die Resource zwecks Load Tracing angesprochen werden kann.

Trace Interval Der Trace Interval gibt die minimale Länge des Intervalls zwischen zwei Trace Meldungen in Sekunden. Null bedeutet, dass für diese Resource keine Trace Meldungen geschrieben werden. Bei jedem Change des Free Amounts wird eine Meldung geschrieben, wenn der Wert 0 dasteht.

Trace Base Der Trace Base gibt die Basislänge des Intervalls über den die Auslastung der Resource im Mittel bestimmt wird. Ist dieser Wert Null, findet keine Berechnung statt. Im Folgenden wird dieser Wert mit D bezeichnet.

Trace Base Multiplier Der Trace Base Multiplier gibt den Strekkungsfaktor (S). Die im Trace protokollierten Zahlen beziehen sich auf : $S^0 * D$, $S^1 * D$ und $S^2 * D$.

Avg. Allocation 1 Gibt die durchschnittliche Belegung der letzten $S^0 * D$ Sekunden.

Avg. Allocation 2 Gibt die durchschnittliche Belegung der letzten $S^1 * D$ Sekunden.

Avg. Allocation 3 Gibt die durchschnittliche Belegung der letzten $S^2 * D$ Sekunden.

Last Write Last Write ist der Zeitpunkt des letzten Schreiben eines Trace Records.

Avg. Allocation Written Gibt die durchschnittliche Belegung seit dem letzten Schreiben eines Trace Records.

13.4.3 Tab Parameters

Der Tab Parameters verwaltet alle im Jobserver bzw. Scope definierten Parameter. Diese Parameter stehen den Jobs und den Programmen, welche innerhalb der Jobs ausgeführt werden, zur Verfügung.
Innerhalb von Scope-Hierarchien findet eine Vererbung der Parameter statt. Es kann ein Parameter auf höchster Ebene definiert werden und jeder Scope und jeder Jobserver, welcher unterhalb dieses angesiedelt ist, hat automatisch diesen Parameter zur Verfügung.
Der Tab Parameters sieht folgendermaßen aus:

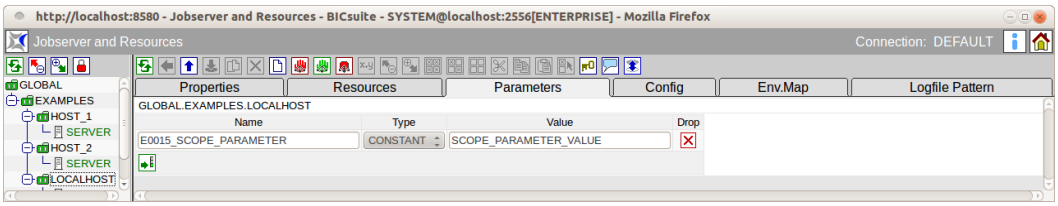


Abbildung 13.10: Jobserver und Scope Parameters

Hier erscheint die Liste aller Parameter. Wurde in der Navigation ein Jobserver angewählt, sind über die aktuellen Parameter hinaus, alle aus der übergeordneten Scope-Hierarchie vererbten Parameter ebenfalls sichtbar. Sie können aber auf dieser Ebene nicht verändert werden. Soll ein vererbter Parameter, auf dieser Ebene abgeändert werden, so muss er mit identischem Namen neu angelegt werden. Dieser Wert überschreibt den Wert aus dem vererbten Teil. Die Anzeige des Wertes ist nun änderbar.
Die Spalten der obigen Liste haben folgende Bedeutung:

Name Hier wird der Name des Parameters angezeigt.

Type Der Typ des Parameters gibt an, um welche Art von Parametern es sich handelt. Es gibt folgende Optionen:

1. Dynamic

   Es handelt sich um einen dynamischen Parameter. Der Wert des Parameters ergibt sich aus dem Wert einer Umgebungsvariablen die auf dem System gepflegt sein muss, auf dem der aktuelle Scope bzw. Jobserver läuft.

   Dieser Wert wird erst zur Startzeit des Jobservers ermittelt. Läuft der Jobserver aktuell auf der Maschine, wird im Feld *Value* der aktuelle Wert der

gleichlautenden Umgebungsvariablen auf der Maschine und der Umgebung in der der Jobserver läuft, angezeigt.

Achtung: Die Umgebungsvariable muss auf der Maschine und in der Startumgebung des Jobservers angelegt worden sein.

2. Constant

   Ist der Typ eines Parameters "Constant", so handelt es sich um einen konstanten Ausdruck, der im Feld *Value* eingetragen werden muss. Verwendet ein Job bzw. dessen Programm den Wert, wird der Parameter während der Ausführung durch diesen konstanten Ausdruck ersetzt.

   Beispiel:

   Wenn ein Skript unabhängig von der aktuellen Datenbank laufen soll, kann die Datenbankverbindung als Parameter abgebildet werden. Wird der Datenbankverbindungsparameter im Scope 1 (zum Beispiel die Test-Datenbank) und gleichzeitig im Scope 2 (zum Beispiel die Produktivdatenbank) ebenfalls mit jeweils unterschiedlichen Werten definiert, kann das Skript ohne Änderungen in beiden Umgebungen laufen, da es seine Datenbankverbindung über den Parameter mitgeteilt bekommt.

Value Hierbei handelt es sich um den Wert des Parameters. Ist der Parameter vom Typ "Constant" muss hier der Parameterwert eingetragen werden. Handelt es sich um einen dynamischen Parameter, wird hier der aktuelle Wert der Umgebungsvariablen auf dem System, auf dem der Jobserver läuft, angezeigt.

13.4.4 Tab Config

Im Tab "Config" steht die Konfiguration des Jobservers beschrieben. Die Konfiguration erfolgt über eine Tabelle mit Key/Value Pairs. Für die Bedeutung der einzelnen Werte wird auf die Jobserver-Dokumentation verwiesen.
Die Felder des Tabs "Config" haben folgende Bedeutung:

Value Die *Value* zeigt den jeweiligen Key/Value Pair an.

Inherit Das Feld *Inherit* zeigt an ob der Key/Value Pair-Wert von einem übergeordneten Scope geerbt wird. Durch setzen des Flags im *Inherit*-Feld wird der aktuelle Eintrag gelöscht. Der Wert des Key/Value Pairs wird in dem Fall über den Vererbungsmechanismus ermittelt. Der Eintrag "DEFAULT" bedeutet, dass es sich um eine Voreinstellung des Systems handelt.

From Das Feld *From* zeigt an, von welchem Scope geerbt wird. Wird als Scope *Default* angezeigt, handelt es sich um Standard Konfigurationswerte.

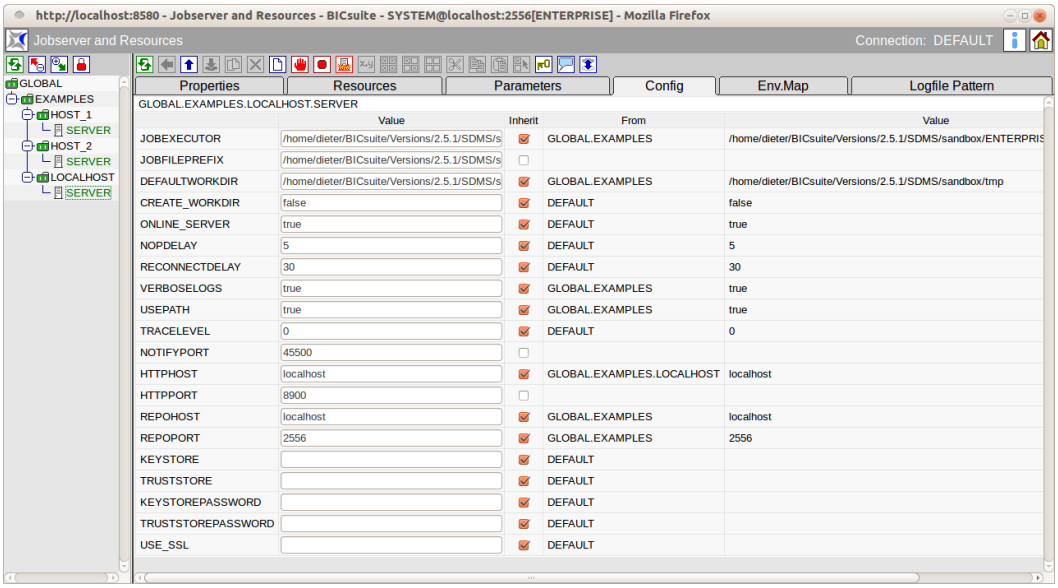


Abbildung 13.11: Jobserver und Scope Konfiguration

Value Der geerbte Wert der Key/Value Pairs.

13.4.4.1 Standard Konfigurationsparameter

Die folgende Tabelle zeigt die Namen der Konfigurationsparameter eines Jobservers, sowie ihre Bedeutung:

Name	**B**edeutung
Name	Bedeutung
JOBEXECUTOR	Der vollqualifizierte Pfad des Jobexecutors
JOBFILEPREFIX	Pfad und Fileprefix der Taskfiles
DEFAULTWORKDIR	Working Directory des Jobservers und default working Directory der über diesen Jobserver ausgeführten Jobs
ONLINE_SERVER	Boolean Wert. Wenn "False" meldet der Jobserver sich ab, wenn gerade keine Arbeit vorliegt
NOPDELAY	Zeit zwischen zwei Anfragen nach Jobs beim Scheduling Server
RECONNECTDELAY	Zeit zwischen zwei Versuchen sich beim Scheduling Server an zu melden

Fortsetzung auf der nächsten Seite

Fortsetzung der vorherigen Seite

Name	**B**edeutung
VERBOSELOGS	Wenn "True" werden zusätzlich zu den Prozessausgaben auch Start- und Endzeit der Jobs in ihren Logfiles geschrieben
USEPATH	Dieser Parameter definiert ob der Jobserver beim Starten von Prozessen die Pfadeinstellung nutzt. Steht "USEPATH" auf "False", müssen alle Programmaufrufe vollqualifiziert sein.
TRACELEVEL	Definiert wie "gesprächig" der Jobserver ist. • 0: Fehler werden protokolliert • 1: Fehler und Warnings werden protokolliert • 2: Fehler, Warnings und Info wird protokolliert • 3: Debug Level; alle Meldungen werden protokolliert
NOTIFYPORT	Der Notifyport ist der Port dem ein UDP-Paket geschickt wird, wenn ein Job für den Jobserver zur Ausführung bereit steht
HTTPHOST	Der Jobserver kann über das HTTP Protokoll Logfiles übermitteln. Dazu müssen beide Parameter "HTTPHOST" und "HTTPPORT" gesetzt sein
HTTPPORT	Port für die Übermittlung von Logfiles
REPOHOST	Hostname oder IP-Adresse des Scheduling Servers
REPOPORT	Port des Scheduling Servers

Tabelle 13.2: Beschreibung der Jobserver Konfigurationsparameters

13.4.5 Tab Env.Map

Beim Starten eines Jobs werden eine Anzahl Standard Parameter des Jobs dem Jobserver als Key/Value Pairs übergeben. Diese Parameter können vor dem Start des Prozesses als Umgebungsvariablen gesetzt werden. Ob und unter welchem Namen diese Variablen sichtbar werden, ist über die Maske "Env.Mapping" konfigurierbar. Die Spalten der obigen Liste haben folgende Bedeutung:

Env.Var. Im Feld *Env.Var.* steht der Name der Umgebungsvariablen.

Sys.Var. Im Feld *Env.Var.* steht der Name des Parameters.

Inherit Das Feld *Inherit* zeigt an ob der Wert des Key/Value Paares von einem übergeordneten Scope geerbt wird. Durch setzen des Flags im *Inherit*-Feld wird

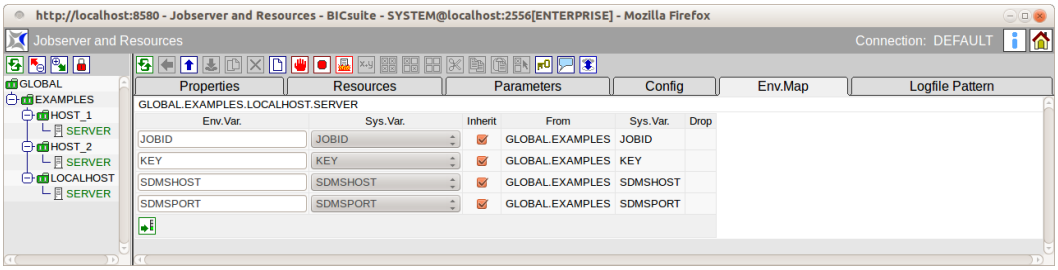


Abbildung 13.12: Jobserver Environment Mapping

der aktuelle Eintrag gelöscht. Der Wert des Key/Value Pairs wird in dem Fall über den Vererbungsmechanismus ermittelt.

From Das Feld *From* zeigt an, von welchem Scope geerbt wird. Wird als Scope *Default* angezeigt, handelt es sich um Standard Konfigurationswerte.

Sys.Var. Im Feld *Env.Var.* steht der Name des Parameters.

13.4.6 Tab Logfile Pattern

Die Jobserver sind in der Lage die Inhalte von Logfiles über das HTTP-Protokoll zu übermitteln. Um zu verhindern, dass beliebige Dateien unkontrolliert gelesen werden, dürfen nur Dateien, deren Name bestimmte Muster aufweisen, angefordert werden.
Zulässige Muster müssen vom Administrator eingetragen werden. Wird dies unterlassen, können keine Dateien angefordert werden.

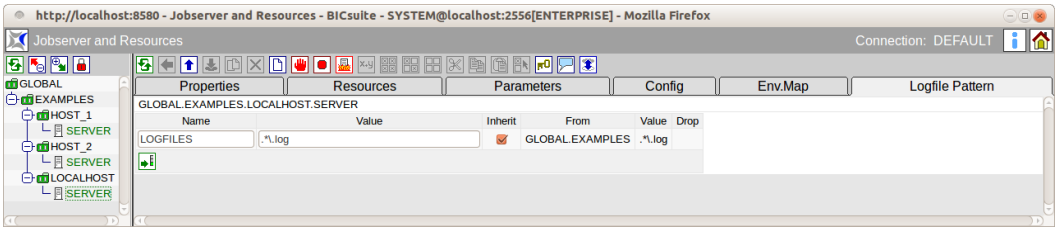


Abbildung 13.13: Jobserver Logfile Name Patterns

Die Logfile Patterns sind Regular Expressions mit einer Besonderheit. Die Zeichenfolge "`/../`" bzw. "`\..\`" (Directory up) darf generell nicht im Namen einer angeforderten Datei auftreten. Dieser Regel vereinfacht es dem Administrator die Logfile Patterns zu definieren.
Generell wird empfohlen den gesamten Namen zu testen. Ein Pattern wie "`.*\.log`" erlaubt es alle Dateien zu lesen, in deren Namen irgendwo die Zeichenfolge "`.log`"

auftritt. Dagegen führt das Pattern "`.*\.log$`" bereits zu der wichtigen Einschränkung, dass der Name auf "`.log`" enden muss.
Im Screenshot wird das Muster "`^/tmp/.*\.log$`" benutzt. Alle Dateien unterhalb des Verzeichnisses `/tmp` die auf `.log` enden, dürfen angefordert werden.
Das Anfordern von Logfiles die keiner der Muster entsprechen, werden als Fehler im Logfile des Jobservers protokolliert. Die Meldungen sehen etwa folgendermaßen aus:

```
ERROR [Jobserver]  01-12-2009 16:43:43 CET [HttpThread] ERROR: Illegal file request : /etc/passwd
```

mit als zusätzlicher Information (die bei jedem File-Request protokolliert wird):

```
...[HttpThread] Got Request from 1.2.3.4 : GET /?FNAME=/etc/passwd HTTP/1.1
...[HttpThread] Got Request from 1.2.3.4 : Host: localhost:8881
...[HttpThread] Got Request from 1.2.3.4 : User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.0.15)
     Gecko/2009102815 Ubuntu/9.04 (jaunty) Firefox/3.0.15
...[HttpThread] Got Request from 1.2.3.4 : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
...[HttpThread] Got Request from 1.2.3.4 : Accept-Language: en-us,en;q=0.5
...[HttpThread] Got Request from 1.2.3.4 : Accept-Encoding: gzip,deflate
...[HttpThread] Got Request from 1.2.3.4 : Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
...[HttpThread] Got Request from 1.2.3.4 : Keep-Alive: 300
...[HttpThread] Got Request from 1.2.3.4 : Connection: keep-alive
...[HttpThread] Got Request from 1.2.3.4 : Cookie: _ZopeId="55722729A4JSGHnrljM"; tree-s="..."
...[HttpThread] Got Request from 1.2.3.4 : Cache-Control: max-age=0
```

Der Anwender bekommt eine Fehlermeldung angezeigt:

```
ERROR: The requested filename doesn't match any of the configured patterns
```

13.5 Pooled Resources

Pooled Resources sind Resources die über den Pool verwaltet werden.

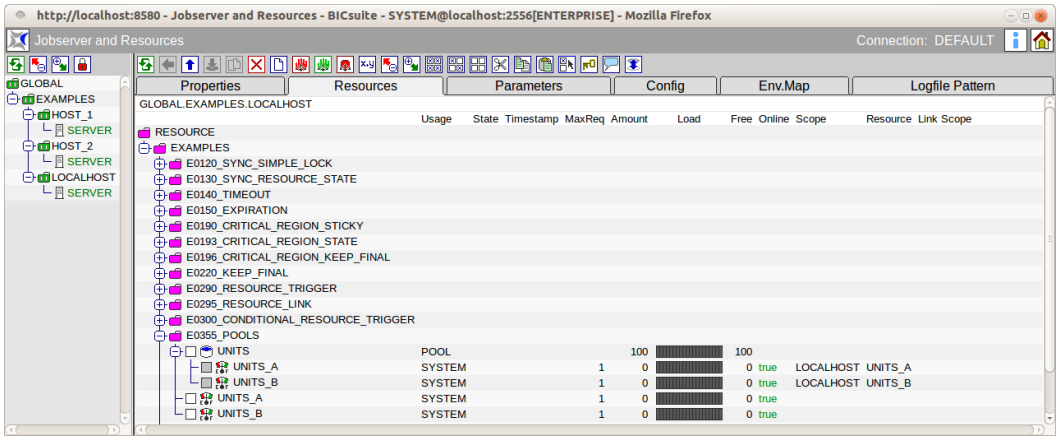


Abbildung 13.14: Pooled Resources

13.5.1 Tab Resource Detail

Der Tab Resource Details wird angezeigt, sobald eine Resource ausgewählt wurde. Der Tab dient zur Eingabe und Pflege der Instanzinformationen für diese Named Resource. Mit Instanz ist hier die Ausprägung der Named Resource im gewählten Scope bzw. Jobserver gemeint. Alle Instanzparameter können hier eingegeben und gepflegt werden.

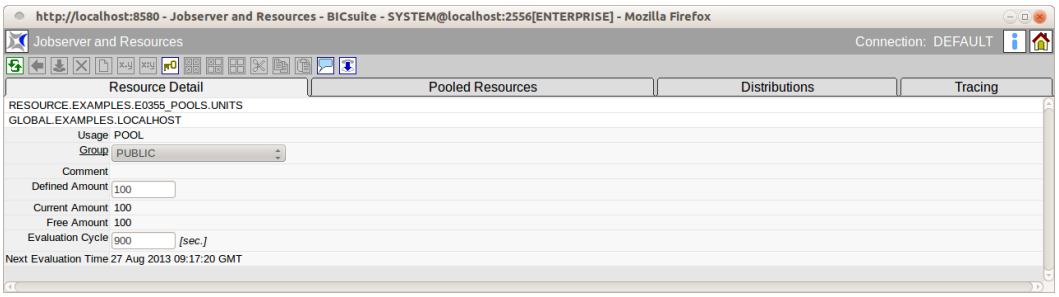


Abbildung 13.15: Pool Details

Usage Die Usage gibt an um welchen Resource Typ es sich handelt.

Group Die Group Option wird benutzt um die Owner-Gruppe auf den spezifizierten Wert zu setzen. Der Benutzer muss zu dieser Gruppe gehören, es sei denn er gehört zu der privilegierten Gruppe ADMIN, in diesem Fall kann jede beliebige Gruppe spezifiziert werden.

Defined Amount Das ist die Menge falls der Pool nicht managed ist.

Current Amount Die aktuelle Menge die zur Verfügung steht.

Free Amount Hierbei handelt es sich um die aktuelle freie Menge.

Evaluation Cycle Das ist der Zeitabstand in Sekunden in dem eine neue Auswertung der Targetamounts stattfindet.

Next Evaluation Time Das ist die Zeit zu der die nächste Auswertung der Targetamounts stattfinden soll.

13.5.2 Tab Pooled Resources

Im Tab *Pooled Resources* kann eine Distribution aktiviert und die aktuelle Situation angezeigt werden.

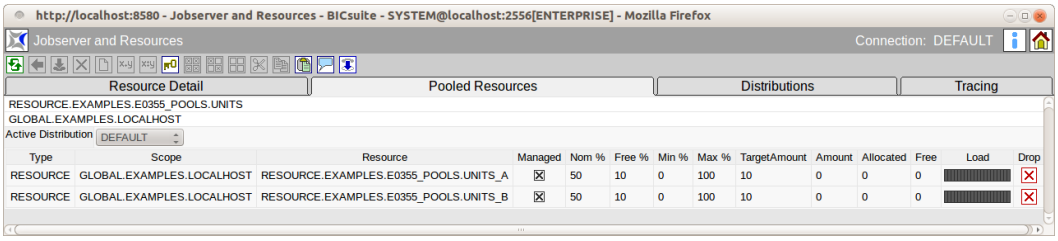


Abbildung 13.16: Pooled Resource; aktuelle Distribution

Active Distribution In dem Auswahlfeld *Active Distribution* wird die aktuelle Distribution eingegeben. Die Standard Distribution wird unter dem Namen *Default* aufgeführt.

Type Hierbei handelt es sich um den Typ des pooled Objektes.

Scope Hierbei handelt es sich um den Scope in dem der Pool angelegt ist.

Resource Hierbei handelt es sich um den vollqualifizierten Namen der Resource.

Managed Ist eine Resource *managed*, wird sie aus dem Pool bedient und es müssen alle anderen Parameter spezifiziert werden. Ist sie *not managed* werden alle übrigen Parameter auf [0] gesetzt, unabhängig davon ob die Parameter im Statement spezifiziert wurden oder nicht.

Nom% Der *Nom%* ist der Nominalwert für die Menge der Resource, ausgedrückt in Prozent.

Free% Der *Free%* ist die Menge der Resource, die idealerweise frei ist, ausgedrückt in Prozent.

Min% Der *Min%* ist die minimale Menge der Resource, ausgedrückt in Prozent.

Max% Der *Max%* ist die maximale Menge der Resource, ausgedrückt in Prozent.

Target Amount Der *Target Amount* ist die aktuelle Sollmenge.

Amount Hierbei handelt es sich um die von der pooled Resource aktuell gehaltene Menge.

Allocated Hier wird die Menge angezeigt, die bereits belegt wurde.

Free Hier wird die freie Menge angezeigt.

Load In diesem Feld wird die graphische Auslastung der Resource angezeigt.

Drop Mit dem *Drop*-Button kann das betreffende Objekt gelöscht werden.

13.5.3 Tab Distributions

Im Tab *Distributions* können Distributions definiert, geändert, kopiert und gelöscht werden.

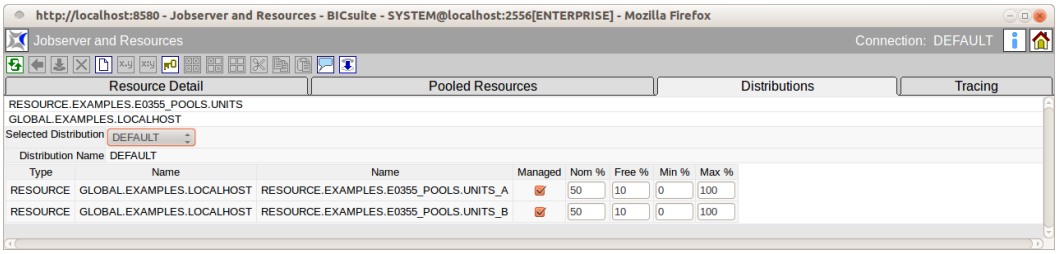


Abbildung 13.17: Pooled Resource; Distributionen

Selected Distribution Im Auswahlfeld *Selected Distribution* kann ein Name ausgesucht werden.

Distribution Name Hier kann ein Name, außer bei Default, vergeben werden.

Type Hierbei handelt es sich um den Typ des pooled Objektes.

Name Hier steht der Name des Scopes in dem die Pooled Resource definiert ist.

Name Hierbei handelt es sich um den vollqualifizierten Namen der Resource.

Managed Hier kann man eingeben ob die Resource verwaltet wird oder nicht.

Min% Im Feld *Min%* wird die minimale Menge der Resource in Prozent eingegeben.

Nom% Im Feld *Nom%* wird der Nominalwert für die Menge der Resource in Prozent eingegeben.

Max% Im Feld *Max%* wird die maximale Menge der Resource in Prozent eingegeben.

Free% Im Feld *Free%* wird die freie Menge in Prozent eingegeben.
Für eine Ausführliche Beschreibung wird auf die Syntax Dokumentation verwiesen.

13.5.4 Tab Tracing

Im *Tab Tracing* kann man die Resourcebelegung zurückverfolgen oder sie überwachen.

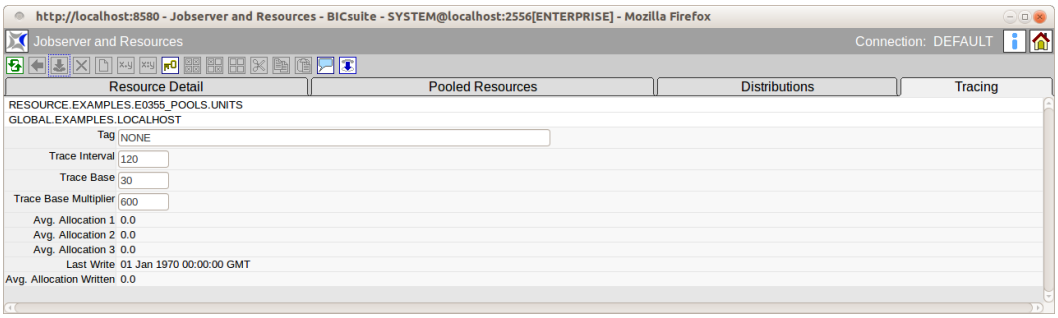


Abbildung 13.18: Pooled Resource; Tracing

Tag Name unter dem die Resource zwecks Load Tracing angesprochen werden kann.

Trace Interval Der Trace Interval gibt die minimale Länge des Intervalls zwischen zwei Trace Meldungen in Sekunden. Null bedeutet, dass für diese Resource keine Trace Meldungen geschrieben werden. Bei jedem Change des Free Amounts wird eine Meldung geschrieben, wenn der Wert 0 beträgt.

Trace Base Der Trace Base gibt die Basislänge des Intervalls über den die Auslastung der Resource im Mittel bestimmt wird. Ist dieser Wert Null, findet keine Berechnung statt. Im Folgenden wird dieser Wert mit D bezeichnet.

Trace Base Multiplier Der Trace Base Multiplier gibt den Strekkungsfaktor (S). Die im Trace protokollierten Zahlen beziehen sich auf : $S^0 * D$, $S^1 * D$ und $S^2 * D$.

Avg. Allocation 1 Gibt die durchschnittliche Belegung der letzten $S^0 * D$ Sekunden.

Avg. Allocation 2 Gibt die durchschnittliche Belegung der letzten $S^1 * D$ Sekunden.

Avg. Allocation 3 Gibt die durchschnittliche Belegung der letzten $S^2 * D$ Sekunden.

Last Write Last Write ist der Zeitpunkt des letzten Schreiben eines Trace Records.

Avg. Allocation Written Gibt die durchschnittliche Belegung seit dem letzten Schreiben eines Trace Records.

13.6 Resource Links

Resource Links erlauben die Allokation von Ressourcen in einem anderen Scope als dem, in welchem ein Job ausgeführt wird.
Werden zum Beispiel in einer Client/Server Umgebung Operationen auf einem Datenbankserver ausgeführt, so werden die Ressourcen welche die Datenbankoperation benötigt, nicht auf dem Client Rechner, auf dem der Job ausgeführt wird belegt, sondern auf dem Rechner auf dem der Datenbankserver liegt.
Resource Links erlauben nun die Abbildung einer Ressource eines Scopes in einem anderen Scope.
Ein Resource Link wird erzeugt, in dem eine Ressource mit Usage STATIC, SYSTEM oder SYNCHRONIZING aus einem Scope bzw. Server ins Clipboard kopiert wird und in einem anderen Scope gepastet wird.
Ist beim Paste eine Resource und/oder ein Resource Link im Clipboard wird folgende Maske angezeigt, auf der ausgewählt werden kann, wie die Paste Operation ausgeführt werden soll.

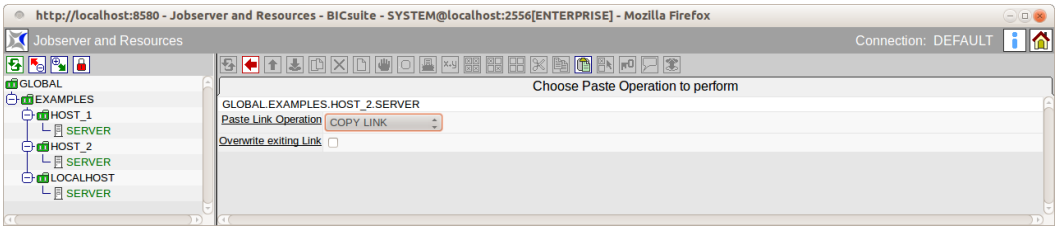


Abbildung 13.19: Resource Links kopieren und erstellen

Ist beim Paste eine Resource im Clipboard so wird das Feld *Paste Resource Operation* angezeigt.
Folgende Auswahlen sind möglich:

- COPY RESOURCE: Die Resource wird kopiert.
- LINK TO RESOURCE: Ein Resource Link wird angelegt.

Ist beim Paste ein Resource Link im Clipboard so wird das Feld *Paste Link Operation* angezeigt.
Folgende Auswahlen sind möglich:

- COPY RESOURCE: Die Resource auf die der Link verweist wird kopiert.
- COPY LINK: Der Link wird kopiert.
- LINK TO LINK: Ein Resource Link der auf den Link verweist wird angelegt.

Ist beim Paste ein Resource Link im Clipboard wird zusätzlich das Feld *Overwite existing Link* angezeigt. Ist diese Checkbox gesetzt, so wird, falls ein Link gleichen Namens bereits existiert, keine Fehlermeldung ausgegeben, sondern der Link überschrieben bzw. geändert.
Resource Links werden in der Ressource Liste mit einem Pfeil im Icon angezeigt wird.

13.6.1 Tab Resource Detail

Der Tab Resource Details eines Resource Links zeigt die Details der über den Resource Link angesprochenen tatsächlichen Ressource. Wird der Resource Link editiert, so werden die Änderungen auf die tatsächlich angesprochene Ressource angewendet.

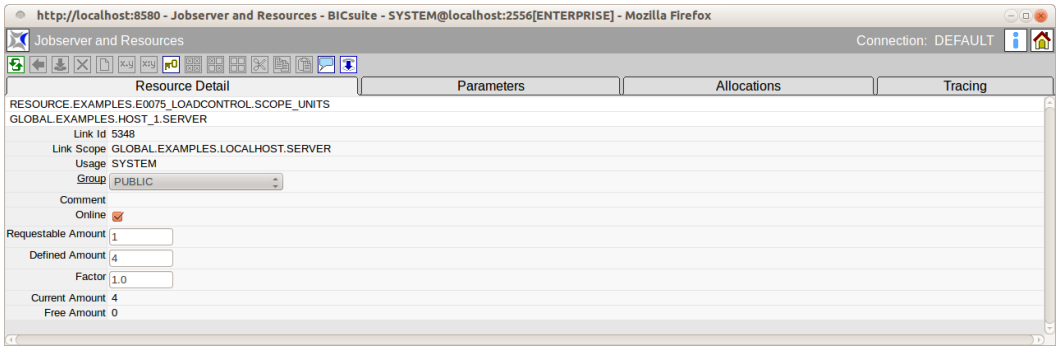


Abbildung 13.20: Resource Link Information

Neben den schon erklärten Attributen der zugrunde liegenden Ressource werden folgende read only Felder angezeigt.

Link Id Die Id der Ressource auf den der Link verweist.

Link Scope Der Scope in dem die verlinkte Ressource liegt.

14 Batches und Jobs

14.1 Bild

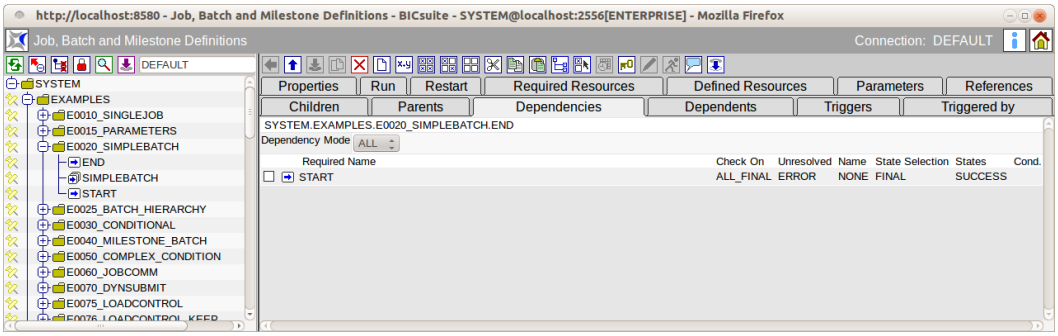


Abbildung 14.1: Batches und Jobs

14.2 Konzept

14.2.1 Kurzbeschreibung

Der Dialog Batches and Jobs dient der Verwaltung aller Job Definitions im BICsuite Scheduling System.
Die Objekte ”Batches”, ”Jobs” und ”Milestones” dienen der Definition der auszuführenden Einheiten des Scheduling Systems.
Ein Job ist die Hülle für ein auszuführendes Programm oder Skript im Scheduling System. Wird ein Job submitted und gestartet, so führt der Jobserver das angegebene Programm oder Skript aus und gibt eine Rückmeldung nach Beendigung des Prozesses über den Erfolg oder Misserfolg (Exit State).
Ein Batch ist ein Container für andere Objekte, er enthält sogenannte Children. Durch Starten des Batches werden automatisch alle enthaltenen Children ebenfalls gestartet. Ein Batch hat kein eigenes Programm oder Skript welches ausgeführt wird.
Der Batch (oder Job), welcher im Dialog Submit Batches and Jobs als erstes Objekt gestartet wird, wird Master Job genannt.
Ein Milestone ist ein Objekt, welches zur Benachrichtigung bei Erreichen (oder Nichterreichen) einer gewissen Anzahl von abgeschlossenen Jobs oder zur Ab-

bildung von komplexen Abhängigkeiten dienen kann. Ein Milestone hat ebenfalls kein lauffähiges Programm oder Skript.

14.2.2 Ausführliche Beschreibung

Jedes Scheduling Entity (Batch, Milestone oder Job) kann in einer Parent-Child-Hierarchie in einem Batch-Lauf stehen. Ein Job wird gestartet, wenn sein Parent gestartet wurde. Das bedeutet, beim Submit einer solchen Ausführungsobjekt-Hierarchie, werden alle Jobs von oben nach unten submitted. Dies geschieht jedoch innerhalb einer Transaktion, so dass alle Aktionen, von außen gesehen, gleichzeitig erfolgen.
Jedes Ablaufobjekt (Batch, Milestone oder Job) kann Abhängigkeiten besitzen. Eine Abhängigkeit beschreibt welches Ausführungsobjekt (required Job) vorher gelaufen sein muss. Soll nur ein bestimmter Exit State beachtet werden, kann der abhängige (dependent) Job erst starten, wenn der benötigte (required) Job vollständig gelaufen ist und den richtigen Exit State zurückgemeldet hat.
Hiermit sind Abhängigkeitsmodelle möglich, die Sicherstellen dass wirklich alle Vorarbeiten erfolgreich durchgeführt werden konnten, um einen nachfolgenden Prozess (z. B. einen Report) zu starten.
Jedes Ablaufobjekt kann verschiedene Ressourcen benötigen. Diese können durch die Eingabe des Environments, eines Footprints und durch die Benennung der benötigten Ressourcen festgelegt werden.
Dies stellt sicher, dass die zur Ausführung des Jobs verwendete Laufzeitumgebung, alle benötigten Ressourcen (z. B. genügend Hauptspeicher, genügend CPU Einheiten, richtige Datenbank, benötigte Systemprogramme etc.) zur Verfügung hat.
Darüber hinaus kann mittels Synchronizing Resources sichergestellt werden, dass Prozesse sich nicht gegenseitig behindern oder das Ressourcen sich in bestimmten States befinden und zeitlich aktuell sind. Über die Ressourcen kann darüber hinaus eine Lastverteilung erreicht werden.
Jedes Scheduling Entity kann bestimmte Parameter definieren oder verwenden. Hiermit ist eine Datenübergabe zwischen einzelnen Jobs bzw. aus der Laufzeitumgebung möglich. Die Parameterübergabe kann innerhalb der Parent-Child-Beziehung in beide Richtungen (von Parent zu Child, von Child zu Parent) stattfinden.
Es ist aber ebenfalls möglich, auch auf Parameter von Scheduling Entities zuzugreifen, welche sich außerhalb der Parent-Child-Beziehung befinden. Hiermit ist eine übergreifende Datenübermittlung zwischen allen Scheduling Entities eines Master Jobs möglich. Weitere Informationen zu Parametern finden Sie im Kapitel 14.5.10.
Für ein Ablaufobjekt können beim Eintritt eines bestimmten Ereignisses (zum Beispiel Beendigung des Jobs) und beim Erreichen bestimmter Exit States, Aktionen definiert werden. Diese Aktionen werden Trigger genannt. Ein Trigger definiert welche Aktion beim Eintreten eines bestimmten Ereignisses ausgeführt werden soll. Die Aktion die ausgelöst wird, ist wieder ein Ausführungsobjekt.
Tritt der Fall ein, dass dieser Trigger ausgelöst wird (er feuert), so wird dieser Job

(oder Batch) submitted. Damit kann zum Beispiel eine Benachrichtigung an einen Systemadministrator erfolgen.

14.2.3 Empfohlene Konvention für Batch Objekte

Um die Übersichtlichkeit zu verbessern und das Arbeiten mit Batch Objekten zu vereinfachen, empfehlen wir unseren Anwendern, sich an folgende Konvention zu halten.
Ein Batch Objekt sollte in einem Ordner mit gleichem Namen liegen. Dieser Ordner sollte den Batch und dessen Kinder (Sub Batches und Jobs) enthalten. Sub Batches sollten wiederum in einem eigenen Unterordner liegen. Job Objekte, welche Kinder haben, sollten wie Batches behandelt werden, also auch in eigenen Unterordnern liegen.
Diese Konvention muss nicht zwingend eingehalten werden. Der Aufwand bei der Vergabe zu Zugriffsrechten, dem Deployment, Kopieren und Verschieben von Teilabläufen etc. kann so jedoch deutlich reduziert und die Orientierung erheblich verbessert werden.
Um die Umsetzung dieser Konvention zu erleichtern wurden folgende Funktionserweiterungen in der BICsuite!web Oberfläche umgesetzt:

- Beim Anlegen eines Ordners kann auch gleich ein Batch Objekt in diesem Ordner mit angelegt werden.

- Beim Anlegen eines Batches kann auch gleich ein übergeordneter Ordner für den Batch angelegt werden.

- Beim Umbenennen von Batches oder Ordnern kann auch der gleichnamige Ordner bzw. Batch mit umbenannt werden.

- Beim Erzeugen von Objekten in einem Ordner mit gleichnamigem Batch oder Job kann das neue Objekt gleich als Kind in den Batch bzw. Job aufgenommen werden.

- Beim Duplizieren (Clonen) von Jobs kann der neue Job als Kind in des gleichnamigen Batches bzw. Jobs des Ordners aufgenommen werden.

- Beim Duplizieren (Clonen) von Foldern kann der gleichnamige Batch bzw. Job des neuen Folders automatisch umbenannt werden.

- Beim Duplizieren (Clonen) von Foldern kann der gleichnamige Batch bzw. Job des neuen Folders als Kind in des gleichnamigen Batches bzw. Jobs des übergeordneten Ordners aufgenommen werden.

14.3 Navigator

Im Navigator werden hierarchisch alle Objekte vom Typ ”Batch” , ”Job” und ”Milestone” innerhalb von Container sog. Folders verwaltet. Die Folders dienen zur besseren Übersichtlichkeit der Objekte.
Darüber hinaus können in Ihnen Parameter und ein Default Environment definiert werden. Allen Jobs, welche sich innerhalb eines Folders befinden, stehen alle Parameters des Folders zur Verfügung. Environment vom Folder verhält sich additiv zum Environment eines Jobs.

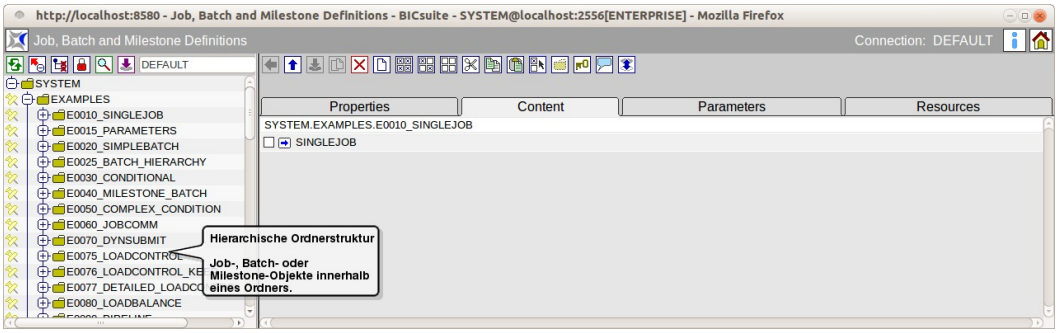


Abbildung 14.2: Batches und Jobs Navigator

14.3.1 Pinning

Um im Folder Navigator eine bessere Übersichtlichkeit zu erreichen, können Objekte durch einen Mausklick auf das *Pin* Icon  ”gepinnt” werden.

Ein Pin kann durch einen Mausklick auf das *Unpin* Icon  wieder entfernt werden.
Zeilen mit einem ”Pin” werden immer angezeigt, unabhängig davon, ob die übergeordneten Folder expandiert sind oder nicht. Der Anwender muss damit im Navigator wesentlich weniger blättern.
Die ”Expand” und ”Collapse” Icons der Baumansicht werden gelb eingefärbt, wenn sich unter einem Folder ein ”gepinntes” Objekt befindet. Damit wird verdeutlicht, dass ein ”Collapse” nicht zum kompletten Schließen des Ordners führt, sondern der Pfad zum ”gepinnten” Objekt immer sichtbar bleibt.

14.3.2 Folder Bookmarks

 Store Bookmark

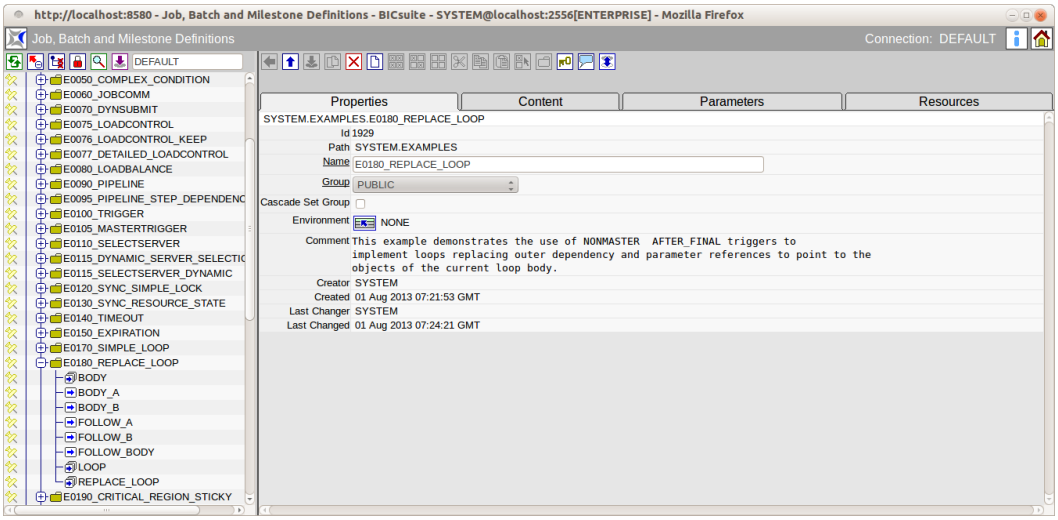


Abbildung 14.3: Batches und Jobs; Ansicht ohne Pin

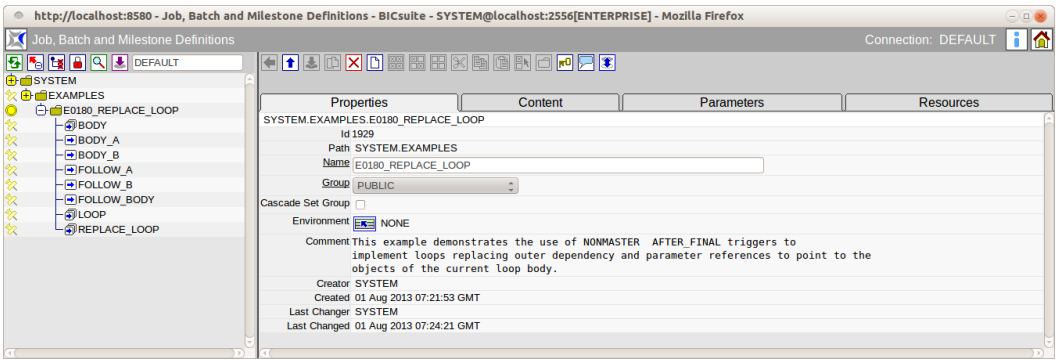


Abbildung 14.4: Batches und Jobs; Ansicht mit Pin

Betätigt man den *Store Bookmark* Button, so wird ein Bookmark für die aktuellen Einstellung des Navigators angelegt.
Die Navigationseinstellungen enthalten die Informationen über Anzeige von Jobs und Batches sowie gesperrte Objekten, den aktuellen Expansions-Status der Baumansicht und ”gepinnte” Objekte.
Der Name unter dem der Bookmark gespeichert werden soll kann im Eingabefeld rechts vom *Store Bookmark* Button eingegeben werden.
Der Bookmark ”DEFAULT” wird beim Aufruf von ”Batches and Jobs” aus dem BICsuite!web Desktop verwendet.
Andere Bookmarks können über Bookmark (Folder) aufgerufen werden.
Falls der aktuelle User ein ”Web GUI Admin” ist, so wird neben dem Namen des Bookmarks noch ein Optionsfeld eingeblendet, über welches eingestellt werden

kann, ob Bookmark als System Bookmark (für alle User Sichtbar) oder als User Bookmark (nur für den aktuellen User sichtbar) gespeichert werden soll.

14.3.3 Folder Search

 Find

Betätigt man den *Find* Button, wechselt der Navigator in den Suchmodus.
Im Eingabefeld kann ein Suchmuster eingegeben werden, welches der, im Optionsfeld rechts vom Eingabefeld, gewählten Syntax entspricht. Unterstützt werden dabei die SQL LIKE Syntax sowie "Regular Expressions".
Über ein weiteres Optionsfeld kann bestimmt werden ob des Muster auf den gesamten Pfad (FULL) oder nur auf das Pfadende (TAIL) angewendet werden soll.
Betätigt man im Suchmodus den *Find* Button, so werden im Navigator die Treffer angezeigt.
Ein Mausklick auf einen Treffer öffnet das Objekt im Editorbereich, das Objekt wird "gepinnt" und der Suchmodus wird wieder verlassen.

14.4 Editor für Folder

14.4.1 Tab Properties

Der Properties-Tab dient zum Erfassen aller Informationen, welche die allgemeinen Eigenschaften des Folders abbilden.

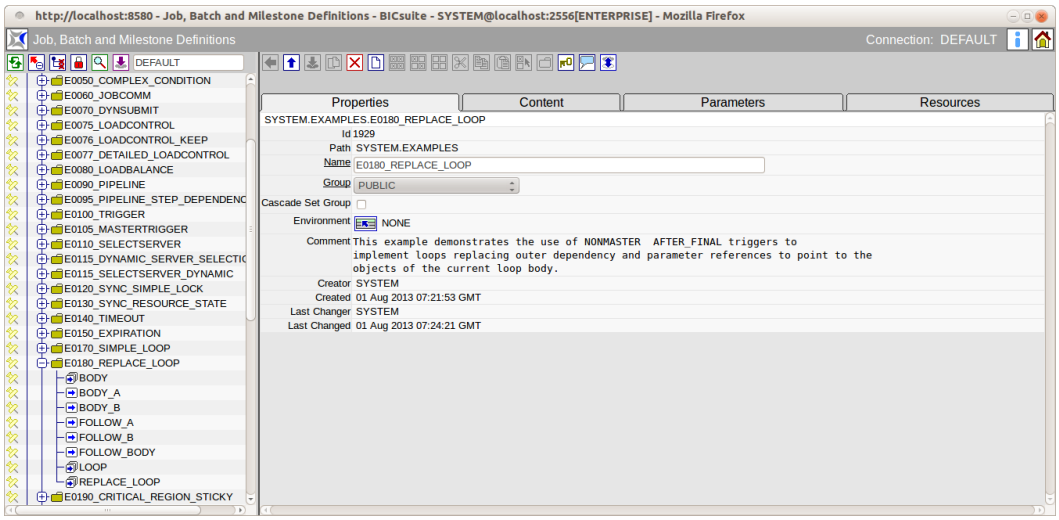


Abbildung 14.5: Folder Properties

ID Die *ID* beschreibt die eindeutige Identifikationsnummer des Objektes.

Path Der *Path* beschreibt die darüberliegende Ordnerhierarchie. Alle übergeordneten Ordner werden punktgetrennt angezeigt.

Name Durch den Namen soll eine verständliche und eindeutige Identifizierung des Objektes im aktuellen Kontext möglich sein. Er muss innerhalb seines Ordners eindeutig sein. Gleichlautende Namen in unterschiedlichen Ordner sind zulässig.

Rename in Content Wird der Name eines Folders geändert und folgt der Folder der "Batch in Folder Konvention" (enthält ein gleichnamiges Batch oder Job Objekt), so wird die Checkbox *Rename in Content* sichtbar. Wird diese gesetzt, so wird beim Umbenennen und Duplizieren (Clonen) des Folders auch das gleichnamige Batch oder Job Objekt im Folder mit umbenannt.

Group Das Feld *Group* gibt die BICsuite Benutzergruppe an, die Eigentümer des Objektes ist.

Cascade Set Group Mit dem *Cascade Set Group* setzt man die Benutzergruppe des Folders und allen darunterliegenden Objekte.

Create Batch in Folder Die Checkbox *Create Batch in Folder* wird bei der Neuanlage eines Folders angezeigt und dient der Unterstützung der empfohlenen Konvention für Folder, Batch und Job Objekte. Ist das Feld angekreuzt, so wird mit dem Folder auch gleich ein Batch Objekt gleichen Namens in dem neu angelegten Folder angelegt.

Batch Exit State Profile Das Feld *Batch Exit State Profile* wird nur angezeigt, wenn die Checkbox *Create Batch in Folder* gesetzt ist und dient der Auswahl des Exit State Profiles für den neu anzulegenden Batch.

Add as Child Die Checkbox *Add as Child* wird in folgenden Fällen angezeigt.
Bei der Neuanlage ist die Checkbox *Create Batch in Folder* gesetzt und der Parent Folder des neu anzulegenden Ordners folgt der Batch in Folder Konvention (enthält ein gleichnamiges Batch oder Job Objekt).
Bei einem bestehenden Folder wird der Name geändert und sowohl der Folder selbst, als auch sein Parent Folder folgen der Batch in Folder Konvention (enthalten ein gleichnamiges Batch oder Job Objekt). In diesem Fall hat die Checkbox *Add as Child* nur eine Auswirkung beim Duplizieren (Clonen) des Folders. Beim Speichern (Umbenennen) des Folders wird die Checkbox *Add as Child* ignoriert.
Mit der Checkbox *Add as Child* kann gewählt werden, ob der neu angelegte Batch (bei Neuanlage) bzw. kopierte Batch bzw. Job (beim Duplizieren bzw. Clonen) auch als Kind des übergeordneten Folder-Batch bzw. -Job eingehängt werden soll.

Environment Das Feld *Environment* ist eine Auswahlbox, in der ein gültiges Environment-Objekt ausgewählt werden kann. Diese Auswahl definiert die zusätzlichen Anforderungen die alle Jobs aus dem Folder an ihre Ablaufumgebung stellen.

Comment Das Feld *Comment* dient als Möglichkeit einer näheren Erläuterung (Kommentar) des Objektes.

14.4.2 Tab Content

Im Tab Content wird der Inhalt des Folders angezeigt.

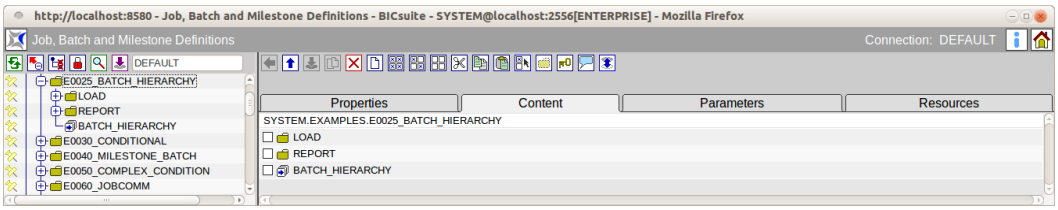


Abbildung 14.6: Folder Content

14.4.3 Tab Parameters

Im Tab Parameters können Parameter definiert werden.

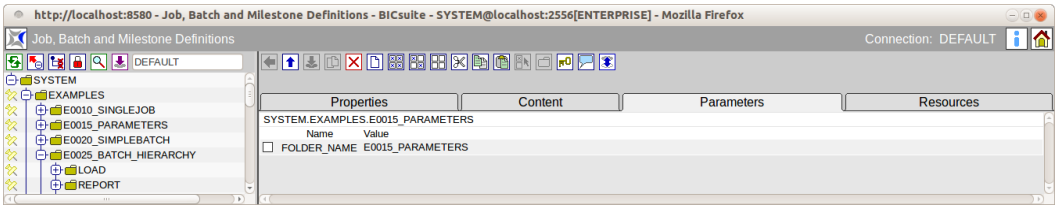


Abbildung 14.7: Folder Parameters

Name Hierbei handelt es sich um den Namen des Parameters. Durch den Namen erfolgt die komplette Werteübergabe der Parameter zwischen einzelnen Jobs.

Value Der *Value* zeigt den Wert des Parameters an.

14.4.4 Tab Resources

Im Tab Resources werden die Ressourcen angezeigt. Durch anklicken des Namens der Resource kommt man in den Tab Resource Details.

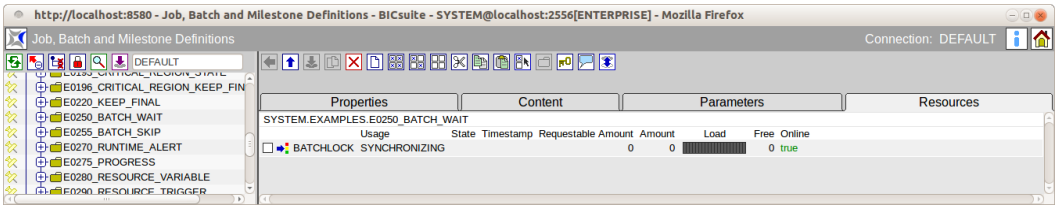


Abbildung 14.8: Folder Resources

Usage Die *Usage* gibt an um welchen Typ "Resource" es sich handelt. Mehr zu Typen von Ressourcen finden Sie im Dialog Named Resource.

State Das Feld *State* ist nur zu sehen, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde.
Hierbei handelt es sich um den aktuellen Status der Resource in diesem Scope oder Jobserver. Der Status kann mittels dieses Wertes gesetzt bzw. geändert werden. In der "Drop Down" Liste sind alle gültigen Resource States des Resource State Profiles enthalten und können ausgewählt werden.
Dies kann bei einer manuellen Fehlerbehebung notwendig sein, um eine Resource wieder in einen Status zu bringen, in dem eine Weiterverarbeitung durch Folgejobs möglich ist.
Falls hier der Wert manuell geändert wird, wird der Wert des Feldes *State* im Tab Resources nicht automatisch aktualisiert. Falls dies gewünscht und notwendig ist, kann dies mittels des *Refresh* Buttons durchgeführt werden.

Timestamp Dieses Feld ist nur zu sehen, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde. Der *Timestamp* gibt die Zeit des letzten Statuswechsels einer Resource an.
Timestamps spielen eine Rolle, falls in einem Ablaufobjekt ein Expiration-Intervall angegeben wurde. Ist dies geschehen, darf der Timestamp nicht älter sein als das angegebene Intervall. Mehr zu Expiration-Zeiten finden Sie im Kapitel 14.5.8.1.
Wird durch die Beendigung eines Jobs eine Änderung des Resource State hervorgerufen, wird der Timestamp automatisch vom BICsuite Server aktualisiert.

Requestable Amount Die Menge der Ressourcen die maximal von einem Job angefordert werden darf.

Amount Beim *Amount* handelt es sich um die Anzahl der vom aktuellen Job belegten oder angeforderten Resource-Instanzen. Übersteigt der Wert des Amounts die aktuell verfügbare Anzahl, die im Feld *Amount* im Tab Resource Detail gepflegt werden können und ist kein alternativer Scope mit einer ausreichenden Anzahl vorhanden, kann der Job nicht ausgeführt werden.

Load Der *Load* zeigt die graphische Auslastung der Resource.

Free Dieses Feld ist nur zu sehen, wenn die Usage der Resource ”Synchronizing” oder ”System” ist.
Der *Free Amount* bezeichnet die Anzahl aller noch nicht von Jobs belegten Instanzen einer Resource, innerhalb des gewählten Scopes oder Jobservers.

Online Mittels *Online* ist es möglich eine Resource in diesem Scope oder Jobserver on- bzw. offline zu schalten. Ist die Resource offline, so steht diese Resource vorübergehend nicht zur Verfügung.

14.5 Editor für Job Definitions

14.5.1 Tab Properties

Der Properties-Tab dient zum Erfassen aller Informationen, welche die allgemeinen Eigenschaften des Job-, Batch-, Milestone- oder Ordnerobjektes abbilden. Je nach Art, des in der Navigation oder während der Neuanlage gewählten Objekttyps, werden im Properties-Tab nur die jeweils notwendigen Informationen angezeigt.

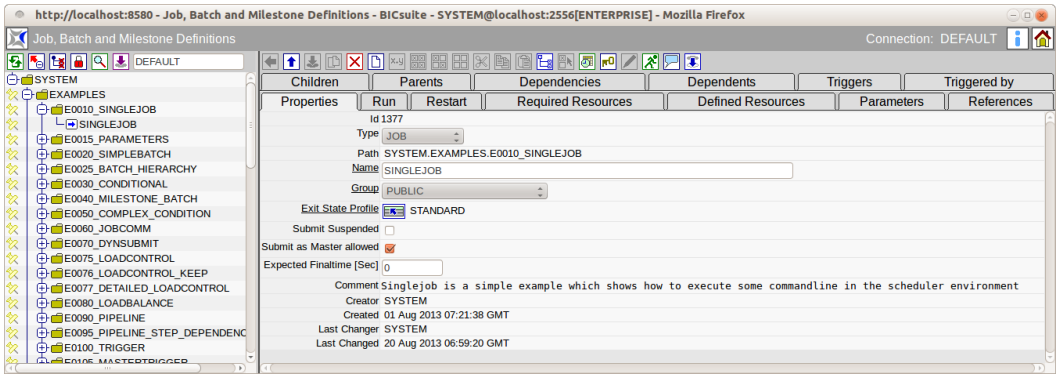


Abbildung 14.9: Job Definition Properties

Die Felder des Tabs ”Properties” haben folgende Bedeutung:

ID Die *ID* beschreibt die eindeutige Identifikationsnummer des Objektes.

Type Der *Type* gibt die Art eines Objektes an. Hier kann aus den folgenden Arten ausgewählt werden:

1. Batch

Der Typ des Objektes ist ein Batch. Ein Batch dient als startbarer Container für eine Anzahl von Jobs, Batches oder Milestones . Er hat selber kein Run Program, dient also nur als Startobjekt seiner Child-Objekte.

Als Beispiel für eine Modellierung als Batch kann eine Anzahl von Jobs dienen, welche täglich laufen müssen. Diese können nun alle unter einem Batch "Täglich" zusammengefasst werden. Alle wöchentlichen Jobs werden ebenfalls unter einem Batch "Wöchentlich" zusammengefasst, darunter auch der Batch täglich, da dieser ja auch zum selben Zeitpunkt als Batch laufen muss. Des weiteren kann ein Batch "Monatlich" erstellt werden, der alle monatlich laufenden Jobs enthält. Somit müssen in diesem System nur diese drei Batches im Time Scheduling angelegt werden und alle Child-Jobs, -Batches bzw. -Milestones werden automatisch zu diesen Zeiten gestartet.

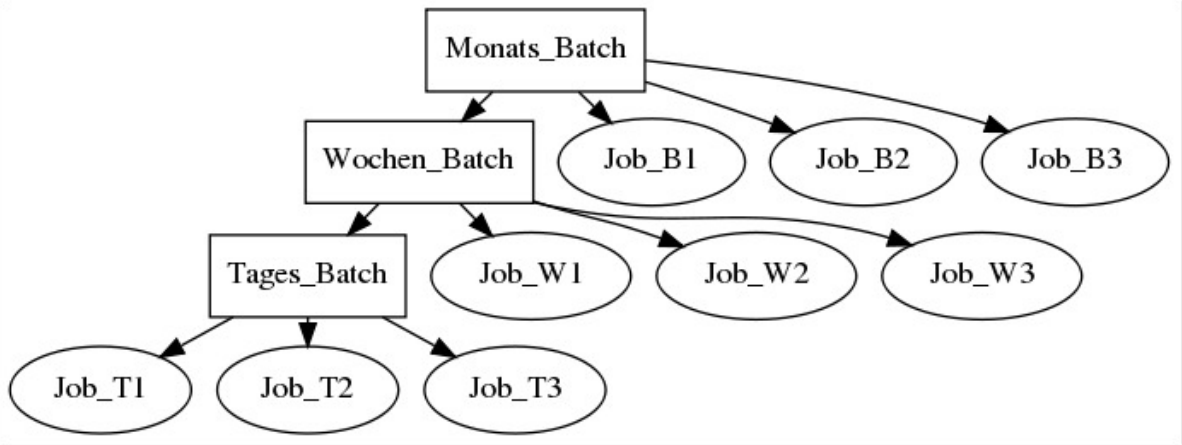


Abbildung 14.10: Batchbeispiele

Die Legende für die Grafik finden Sie im Kapitel 1.8.

2. Milestone

   Ein Milestone wird aus mehreren Gründen benötigt. Erstens dient er als Platzhalter für einen bestimmten Zeitpunkt im Ablauf eines komplexen Batch-Laufes. Wird dieser Milestone erreicht, so kann mittels eines Triggers eine Aktion durchgeführt werden (zum Beispiel eine SMS oder Mail gesendet werden).

   Als zweiten Grund werden Milestones benötigt um komplexe Abhängigkeiten zwischen einzelnen Jobs zu definieren. Da es für einen Job immer nur möglich ist von allen benötigten Objekten oder von mindestens einen benötigten Objekt abhängig zu sein, so ist es nicht möglich komplexere Abhängigkeiten (von Job1 und Job2 oder Job3 etc.) direkt zu definieren.

Diese Möglichkeiten können nun mittels eines Milestones definiert werden. Durch die Modellierung eines Milestone1, der erreicht wird wenn Job1 und Job2 erreicht wurden, kann die Abhängigkeit unseres Jobs von Job3 oder Milestone1 modelliert werden und die Bedingung ist nun abbildbar.

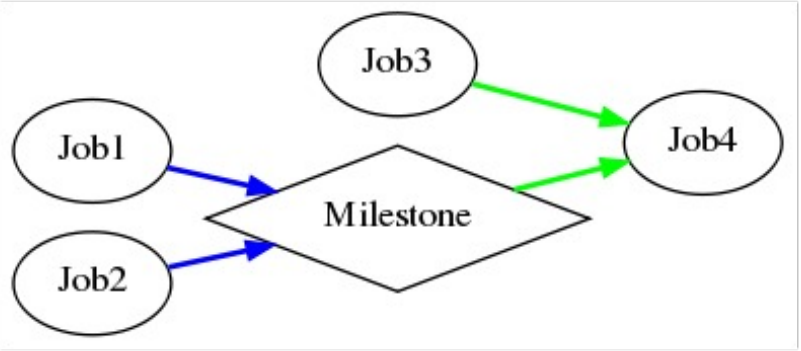


Abbildung 14.11: Beispiel für Milestone

Die Legende für die Grafik finden Sie im Kapitel 1.8.

Milestones haben außerdem eine einzigartige Eigenschaft. Es können für Milestones dynamische Children definiert werden. Jedes Mal wenn ein Job, der auch Child eines Milestones ist, jetzt dynamisch submitted wird, bekommt der Milestone diesen Job ebenfalls als Child.

Diese Funktionalität ermöglicht eine ”Pipeline”-Verarbeitung. Nehmen wir zum Beispiel an, dass Daten einer (partitionierten) Tabelle verarbeitet werden sollen. Aufgrund der Partitionierung wird beschlossen die Arbeit zu parallelisieren. Für jede Partition werden die relevante Daten extrahiert und anschließend in irgendeiner Form parallel weiter verarbeitet. So bald alle Extraktionsjobs fertig sind, soll eine weitere Verarbeitung laufen. Wird nun der Extraktionsjob als dynamisches Child eines Milestones definiert, dann wird der Milestone genau dann den finalen Status erreichen, wenn alle Extraktionsjobs erfolgreich beendet sind. Die weitere Verarbeitung der Tabelle kann abhängig gemacht werden vom Milestone und damit wiederum parallel zu der Verarbeitung der extrahierten Partitionsdaten ablaufen.

3. Job

   Alle ausführbaren Objekte (Programme, Skripte etc.), welche im Scheduling System auf einem Jobserver ausgeführt werden sollen, müssen als Job definiert werden.

4. Folder

Ein Folder stellt einen Ordner innerhalb der Ordnerhierarchie für die Ausführungsobjekte dar. Er dient zur Ordnung der anderen Objekte, da in einem normalen Scheduling System viele verschiedene Jobs Batches und Milestones verwaltet werden und bei einer flachen Hierarchie schnell die Übersicht verloren gehen kann. Der Aufbau der Hierarchie ist dem jeweiligen Benutzer selbst überlassen. Es könnte sich eine logische Ordnung (alle Objekte zum selben Thema etc. in einem Ordner) oder eine Abbildung der Abteilungs- oder Personalstruktur (alle Objekte vom Benutzer/Abteilung X in einem Ordner) anbieten. Es sollte nur möglich sein sich in der Hierarchie schnell zurecht zu finden und die benötigten Objekte einfach zu finden.

Folder können nicht ausgeführt werden und können keine Abhängigkeiten besitzen. Es ist möglich ein Environment für einen Ordner zu definieren. Darüber hinaus ist es möglich Parameter zu definieren, welche alle Jobs die sich innerhalb des Ordners befinden verwenden können.

Path (Pfad) Der *Path* beschreibt die darüberliegende Ordnerhierarchie. Alle übergeordneten Ordner werden punktgetrennt angezeigt.

Name Durch den Namen soll eine verständliche und eindeutige Identifizierung des Objektes im aktuellen Kontext möglich sein. Er muss innerhalb seines Ordners eindeutig sein. Gleichlautende Namen in unterschiedlichen Ordner sind zulässig.

Decomposite Die Checkbox *Decomposite* angezeigt wenn der Typ eines Objektes von Job nach Batch geändert wird. Ist diese gesetzt, so wird eine "Decomposition" ausgeführt. Dabei wird nicht nur der Objekt Typ nach Batch geändert, sondern es wird auch ein Job als Kind dieses Batches angelegt, welcher die Job Properties des ursprünglichen Job Objektes übernimmt. Das nach einer Decomposition entstandene Batch Objekt (mit einem Job als Kind) verhält sich im Kontext des Scheduling System wie das zuvor existierende Batch Objekt. Alle Time Schedules, Abhängigkeiten, Parameter References. etc. bleiben erhalten. Es entsteht eine optimale Ausgangssituation für die Zerlegung des Jobs in kleinere Einzelschritte (Process Decomposition).

Job Name Das *Job Name* wird nur bei gesetzter *Decomposite* Checkbox angezeigt. Hier kann der Name des bei der Decomposition anzulegenden Job Objektes bestimmt werden.

Rename Parent Folder Wird der Name eines Batches geändert und folgt der Batch der "Batch in Folder Konvention" (liegt in einem gleichnamigen Folder), so wird die Checkbox *Rename Parent Folder* sichtbar. Wird diese gesetzt, so wird beim Speichern (Umbenennen) des Batches auch der gleichnamige Parent Folder

mit umbenannt. Bei Duplizieren (Clonen) eines Batches wird die Checkbox *Rename Parent Folder* ignoriert.

Create Batch Folder Die Checkbox *Create Batch Folder* wird angezeigt wenn eine Batch neu angelegt wird oder der Typ eines Objektes nach Batch geändert wird. Sie dient der Unterstützung der empfohlenen Konvention für Folder, Batch und Job Objekte. Ist das Feld angekreuzt, so wird für den Batch ein Folder mit dem Namen des Batches erzeugt und der neue Batch in diesem Folder angelegt. Die Checkbox *Create Batch Folder* wird auch angezeigt, wenn der Typ eines Objektes von Job auf Batch geändert wird (Decomposition).

Add as Child Die Checkbox *Add as Child* wird in folgenden Fällen angezeigt.
Bei der Neuanlage eines Batch, Job oder Milestone Objektes, wenn der Parent Folder des anzulegenden Objektes der Batch in Folder Konvention (enthält ein gleichnamiges Batch oder Job Objekt) folgt.
Bei einem bestehenden Job oder Milestone wird der Name geändert und der Parent Folder folgt der Batch in Folder Konvention. In diesem Fall hat die Checkbox *Add as Child* nur eine Auswirkung beim Duplizieren (Clonen) des Jobs oder Milestones. Beim Speichern (Umbenennen) wird die Checkbox *Add as Child* ignoriert.
Mit der Checkbox *Add as Child* kann gewählt werden, ob das neu angelegte Objekt (bei Neuanlage) bzw. kopierte Job bzw. Milestone (beim Duplizieren bzw. Clonen) auch als Kind des übergeordneten Folder-Batch bzw. -Job eingehängt werden soll.

Group Das Feld *Group* gibt die BICsuite Benutzergruppe an, die Eigentümer des Objektes ist.

Exit State Profile In diesem Auswahlfeld kann ein Exit State Profile ausgewählt werden.
Der Schalter *Submit Suspended* macht es Möglich den tatsächlichen Start eines Ablaufes zu verzögern. Zum Beispiel könnte für einen Job eine externe Genehmigung durch eine dritte Person nötig sein. Erst wenn diese erfolgt ist, darf der Job starten.

Submit Suspended Der Schalter *Submit Suspended* gibt die Möglichkeit den tatsächlichen Start eines Ablaufes zu verzögern. Zum Beispiel könnte für ein Job eine externe Genehmigung durch eine dritte Person nötig sein. Erst wenn diese erfolgt ist, darf der Job starten.

Resume Wird nur angezeigt falls *Submit Suspended* gesetzt wurde.
Hier kann ausgewählt werden, ob ein automatischer Resume stattfinden soll.
Es gibt folgende Möglichkeiten:

- NO: wählt diese Funktionalität ab.

- AT: wählt einen automatischen Resume zu einem festen Zeitpunkt. Das Eingabefeld *Resume Time* wird angezeigt.

- IN: wählt einen automatischen Resume nach Ablauf einer Zeit. Die Eingabefelder *Resume In* und *Unit* werden angezeigt.

Resume Time Wird nur angezeigt falls im *Resume* "AT" gewählt wurde. Hier wird der gewünschte Resume Zeitpunkt im Format "YYYY-MM-DDTHH:MI:SS" eingegeben.
Das Format orientiert sich an der ISO Norm 8601 und erlaubt auch unvollständige Angaben. Die Eingabe von 'T16:00' wird den Job um 16:00 Uhr resumen (ausgehend von der aktuellen Zeit).

Resume In Wird nur angezeigt falls im *Resume* "IN" gewählt wurde.
Hier wird angegeben, wie viele Zeiteinheiten (siehe *Unit*) bis zum Resume gewartet werden soll.

Unit Wird nur angezeigt falls im *Resume* "IN" gewählt wurde.
Hier wird eingegeben, ob es sich bei der Eingabe im *Resume In* um Minuten (MINUTE), Stunden (HOUR), oder Tage (DAY) handeln soll.

Nice Value Das Feld ist nur sichtbar falls es sich bei dem Objekt um einen Batch handelt.
Der *Nice Value* gibt an, mit welcher Priorität Children dieses Objektes laufen sollen. Der Nice Value gibt einen Offset an, der auf die Priorität der Children addiert wird.
Beispiel:
Besitzt der Prozess einen Nice Value von -50 und das Child hat eine Priorität von 100, dann ist die Priorität des Child-Prozesses des Batches 50. Würde der Child-Prozess alleine gestartet wäre die Priorität 100.

Submit as Master Allowed Dieses Feld ist nur sichtbar falls es sich bei dem Objekt um einen Batch oder einen Job handelt.
Dieser Schalter gibt an ob es möglich ist diesen Job als Master Job im Dialog Submit Batches and Jobs zu submitten. Ist der Schalter nicht gesetzt, so ist dies nicht möglich und der Job oder Batch kann nicht eigenständig, sondern nur als Child von irgendeinem übergeordneten Job ausgeführt werden.

Comment Das Feld *Comment* dient als Möglichkeit einer näheren Erläuterung (Kommentar) des Objektes.

14.5.2 Tab Run

Der Tab Run erscheint nur, falls in der Navigation ein Objekt vom Typ ”Job” handelt oder im Falle einer Neuanlage der Typ ”Job” in der Auswahlbox gewählt wurde.
Im Tab Run werden alle Informationen über die auszuführenden Kommandozeilen eines Jobobjektes verwaltet.

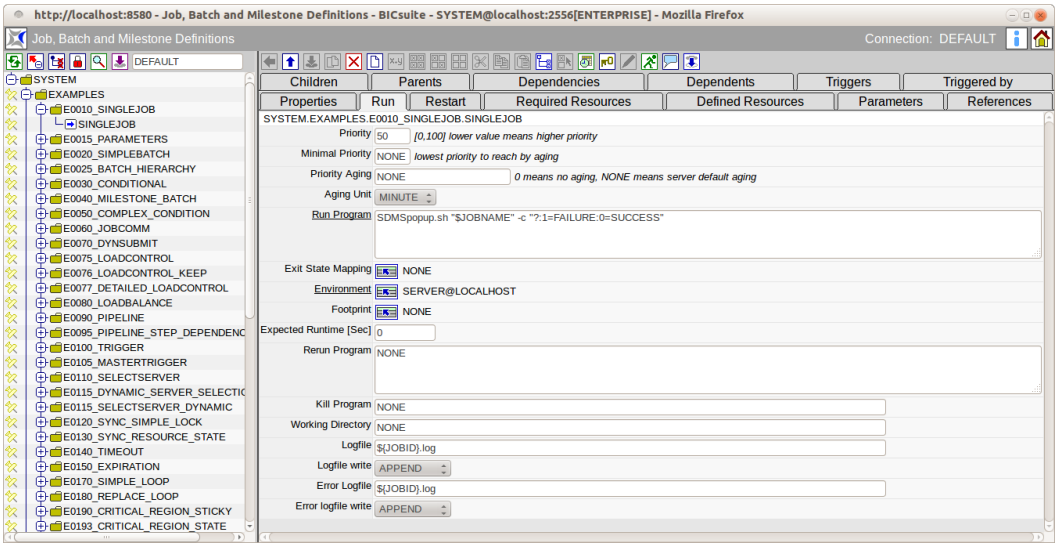


Abbildung 14.12: Job Run Tab

Die Felder des Tabs ”Run” haben folgende Bedeutung:

Priority Das Feld *Priority* gibt an, mit welcher Dringlichkeit der Prozess, falls er gestartet werden soll, vom Scheduling System berücksichtigt wird. Der Wertebereich der Priorität erstreckt sich von 100 (sehr niedrig) bis 0 (sehr hoch). Im System werden alle startbaren Jobs mit ihren Prioritäten berücksichtigt und dann der Job mit der höchsten Priorität gestartet. Haben zwei Jobs eine identische Priorität wird der Job mit niedrigster ID gestartet.
Mit der Länge der Wartezeit eines Jobs, der darauf wartet im Scheduling System gestartet zu werden, erhöht sich seine Priorität. Das heißt, wird ein Prozess mit einer Priorität von 50 gestartet, so steigt seine Priorität pro Zeiteinheit. Dies ist im Scheduling System in der Konfiguration einstellbar. Zum Beispiel kann ein Job pro halbe Stunde um einen Prioritätspunkt steigen. Das heißt, nach einer halben Stunde ist die Priorität bei 49, nach einer Stunde bei 48 usw. Damit steigen ”ältere” Jobs in der Priorität und werden vor neu submitteten Jobs berücksichtigt.

Minimal Priority Minimale effektive Priorität die durch das natürliche Altern erreicht werden kann.

Priority Aging Die Anzahl Zeiteinheiten nach der die effektive Priorität um 1 erhöht wird.

Aging Unit Die Zeiteinheit die für das Alterungsintervall genutzt wird.

Run Program Im Feld *Run Program* wird eine Kommandozeile angegeben, die von einem geeigneten Jobserver ausgeführt werden soll. Die Kommandozeile zuerst vom Scheduling System interpretiert und in Programmaufruf und einzelne Parameter zerlegt. Dabei werden die Regeln der Bourne-Shell in Bezug aufs Quoting befolgt.
Sollen Environment-Variablen oder Parameter angegeben werden, so müssen diese konform der jeweiligen Ausführungsumgebung (je nach Shell oder Kommandozeileninterpreter der Umgebung, siehe Environment) gequoted werden. Eine Auflistung aller verwendbaren Standardparameter finden Sie in Paragraph 14.5.10.2.
Für weitergehende Informationen über das zu verwendende Quoting, lesen Sie bitte die Dokumentation Ihres verwendeten Kommandozeileninterpreters (Shell).
Falls es sich beim Zielsystem um Windows handelt, muss mit einigen weiteren Schwierigkeiten gerechnet werden. Wenn unter Windows eine Tabelle aus Executable und Parameter zur Ausführung angeboten wird, konstruiert Windows zuerst eine Kommandozeile aus der Tabelle, um diese Kommandozeile anschließend wieder zu interpretieren. Problematisch ist auch die Verwendung von Quoting unter Windows, da die aufgerufene Programme die Quotes entfernen oder eben nicht entfernen, je nach Typ des Executables (`.exe` oder `.bat`).
Ist das auszuführenden Programm (erstes Element der Kommandozeile) eine gültige Ganzzahl, so wird die Kommandozeile nicht vom Jobserver ausgeführt, sondern der Job so behandelt, als hätte er sich mit der Ganzzahl als Exit Code beendet. Dummy Jobs mit 'true' oder 'false' als Programm können nun als '0' statt 'true' bzw. '1' statt 'false' implementiert und so vom System wesentlich effizienter und schneller verarbeitet werden.
Sollte es tatsächlich einmal nötig sein ein Executable mit einer Zahl als Namen auszuführen, so kann dies durch einen Pfad Prefix ('./42' statt '42') erreicht werden.

Exit State Mapping Im Feld *Exit State Mapping* kann ein bestehendes Exit State Mapping eingegeben werden. Dieses Mapping definiert die Abbildung des Exit Codes eines Jobs nach einem Exit State. Wird kein Mapping angegeben, wird das default Mapping aus dem Exit State Profile genutzt. Das Mapping muss mit dem Profile verträglich sein.

Environment Im Feld *Environment* muss ein bestehendes Environment ausgewählt werden. Alle Named Resources dieses Environments müssen auf einem Jobserver vorhanden sein, damit der Job auf diesem Jobserver ausgeführt werden kann.

Footprint Im Feld *Footprint* kann ein bestehender Footprint gewählt werden. Alle Named Resources für diesen Footprint müssen auf einem Jobserver in ausreichender Anzahl vorhanden sein, damit der Job auf diesem Jobserver ausgeführt werden kann.

Expected Runtime Die *Expected Runtime* beschreibt die erwartete Zeit, die ein Job für seine Ausführung benötigt. Sie ist manuell einzutragen und kann z. B. für Laufzeitüberwachung genutzt werden.

Expected Finaltime Die *Expected Finaltime* beschreibt die erwartete Zeit, die ein Job für seine Ausführung benötigt (von Submit bis zum Erreichen eines finalen Status). Sie ist manuell einzutragen und wird für die Darstellung im Kalender verwendet. Natürlich gibt es auch einen Standard Parameter dazu, so dass der erwartete Wert dem Job zur Verfügung steht.

Rerun Program Das Feld *Rerun Program* gibt das Kommando an, welches bei einer wiederholten Ausführung des Jobs nach einem Fehlerzustand (Rerun) ausgeführt werden soll. Dabei kann es sich um ein anderes Kommando handeln als das Run Program, da es vielleicht nötig ist bei einem wiederholten Start Aufräumarbeiten auszuführen, bevor das ursprüngliche Kommando wieder aufgesetzt werden kann. Wurde im Rerun Program kein Wert eingetragen, so wird bei einem wiederholten Start des Jobs das Run Program aufgerufen.
Zu Details bzgl. der Verarbeitung (Quoting, Behandlung von Integern als Program Name) siehe 'Run Program' weiter oben.

Kill Program Das *Kill Program* bestimmt welches Programm ausgeführt werden soll, um einen aktuell laufenden Job zu beenden.
Ist in diesem Feld kein Wert eingetragen, so ist es nicht möglich den Job über die BICsuite Oberfläche vorzeitig zu terminieren.
Ein Beispiel für ein Kill Program könnte in einer Unix-Umgebung das **"kill"** Kommando sein. Im Kill Program kann auch über die vordefinierte Systemvariable "PID" auf die Process Id des laufenden Jobs zugegriffen werden (Beispiel: "kill -9 $PID").

Working Directory Das *Working Directory* gibt an, in welchem Verzeichnis das Run Program gestartet werden soll. Der Jobserver wechselt vor dem Starten des Run Programs in das angegebene Verzeichnis.
Wurde in diesem Feld kein Wert eingegeben, so wird als Working Directory das Default Directory des Jobservers verwendet.
Es ist auch möglich als Working Directory einen Parameter, Folder-, Scope- oder Jobserver-Parameter anzugeben.

Logfile Das Feld *Logfile* gibt an, in welche Datei alle normalen Ausgaben des Run Programs ausgegeben werden sollen. Unter normalen Ausgaben sind alle Ausgaben gemeint die den normalen Ausgabekanal (STDOUT unter UNIX) benutzen.
Auch im Feld *Logfile* sind Parameterersetzungen möglich. So kann zum Beispiel der Parameter JOBID zur eindeutigen Kennzeichnung eines Jobs oder ein TIMESTAMP zur Kennzeichnung des Datums verwendet werden.
Wurde kein Wert eingetragen, so werden alle Ausgaben verworfen.
Wurde im Logfile keine Pfadangabe angegeben, so wird das Logfile im Working Directory geschrieben.

Logfile Write Mit dem Feld *Logfile Write* wird dem System mitgeteilt wie ein Logfile anzulegen ist. Es gibt folgende Optionen:

- Append

  Mit Append werden alle Ausgaben des Logfiles an ein bestehendes Logfile angehängt. Falls noch kein Logfile existiert, wird ein neues erzeugt.

- Truncate

  Mit Truncate wird bei jedem neuen Ablauf das Logfile gelöscht und neu angelegt.

Error Logfile Das Feld *Error Logfile* gibt an, in welcher Datei alle Fehlerausgaben des Run Program ausgegeben werden sollen. Unter Fehlerausgaben sind alle Ausgaben gemeint, die den Fehlerausgabekanal (stderr unter UNIX) benutzen.
Auch im Feld *Error Logfile* sind Parameterersetzungen möglich. So kann zum Beispiel der Parameter JOBID zur eindeutigen Kennzeichnung eines Jobs oder ein TIMESTAMP zur Kennzeichnung des Datums verwendet werden.
Wurde kein Wert eingetragen, so werden alle Ausgaben verworfen.
Wurde im Logfile keine Pfadangabe angegeben, so wird das Logfile im Working Directory geschrieben.
Wird derselbe Name, wie im Feld *Logfile* angegeben, so werden diese Ausgaben ebenfalls in diese Datei geschrieben. Die Reihenfolge der Ausgaben (Error oder Normal) sind allerdings durch die Pufferung zufällig. Es werden jedoch unabhängig vom Inhalt des Feldes Error Logfile Write keine Ausgaben des normalen Ausgabekanals (stdout) überschrieben.

Error Logfile Write Mit dem Feld *Error Logfile Write*, wird dem System mitgeteilt, wie das Error Logfile anzulegen ist. Es gibt folgende Optionen:
ERROR LOGFILE WRITE OPTIONEN

- Append

Mit Append werden alle Ausgaben des Error Logfiles an ein bestehendes Error Logfile angehängt. Falls noch kein Error Logfile existiert, so wird ein neues erzeugt.

- Truncate

  Mit Truncate wird bei jedem neuen Ablauf das Error Logfile gelöscht und neu angelegt.

14.5.3 Tab Restart

Im Tab Restart kann das Restartverhalten eines Jobs definiert werden.
Ein Job wird nur neu gestartet, wenn er sich im Job Status ”FINISHED” befindet.
Erhält ein Job in einen ”RESTARTABLE” Exit Status aufgrund eines Job Status ”ERROR”, so wird KEIN Restart durchgeführt!

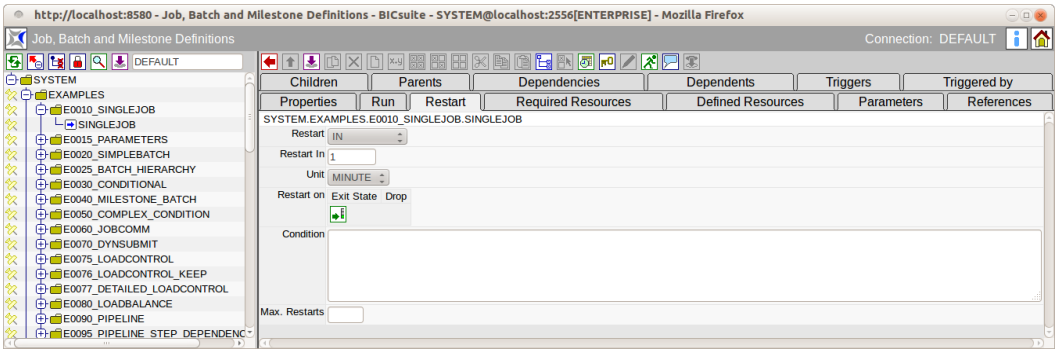


Abbildung 14.13: Job Tab Restart

Die Felder des Tabs ”Restart” haben folgende Bedeutung:

Restart Bestimmt ob der Job automatisch neu gestartet werden soll, falls er einen ”RESTARTABLE” Exit Status erreicht.
Folgende Auswahlen sind möglich:

- NO: Kein Restart. Auf dem Tab ”Restart” werden keine weiteren Felder angezeigt.
- IMMEDIATE: Sofortiger Neustart.
- AT: Neustart zu einem gegebenen Zeitpunkt (z.B. ”T16:00” um 16:00 Uhr)
- IN: Neustart nach einer Wartezeit (z.B. 10 Minutes)

Restart Time Wird nur angezeigt wenn im Feld *Restart* "AT" gewählt wurde.
Zeit zu der der Job neu gestartet werden soll im Format "YYYY-MM-DDTHH:MI:SS". Das Format orientiert sich an der ISO Norm 8601 und erlaubt auch unvollständige Angaben. Die Eingabe von 'T16:00' wird den Job "suspended" neu starten und um 16:00 Uhr resumen (ausgehend vom Submit Zeitpunktes des Jobs).

Restart In Wird nur angezeigt wenn im Feld *Restart* "IN" gewählt wurde.
Anzahl Einheiten (siehe *Unit*) die gewartet werden soll bis der Job wieder anläuft.

Unit Wird nur angezeigt wenn im Feld *Restart* "IN" gewählt wurde.
Einheiten für die Wartezeit (siehe *Restart In*).

Restart On Tabelle der Exit Status welche zu einem Restart führen sollen.
Ist die Tabelle leer so wird bei allen "RESTARTABLE" Exit Status nur gestartet.

Condition Optionale Bedingung (ab BICsuite PROFESSIONAL) für das Auslösen einen Restarts.

Max. Restarts Maximale Anzahl, wie oft der Job neu gestartet wird.
Wird nichts eingegeben gilt der Systemparameter TriggerSoftLimit. Die Eingabe von 0 startet den Job beliebig oft wieder.

14.5.4 Tab Children

Im Tab Children werden alle Ausführungsobjekte als Liste angezeigt, welche als Child des aktuellen Jobs definiert worden sind. Mittels Copy and Paste können weitere Objekte als Children hinzugefügt werden.
Eine Reihenfolge der Ausführung der Children ist nicht festgelegt, sie kann aber über die Dependencies festgelegt werden.
Der Tab Children sieht folgendermaßen aus:

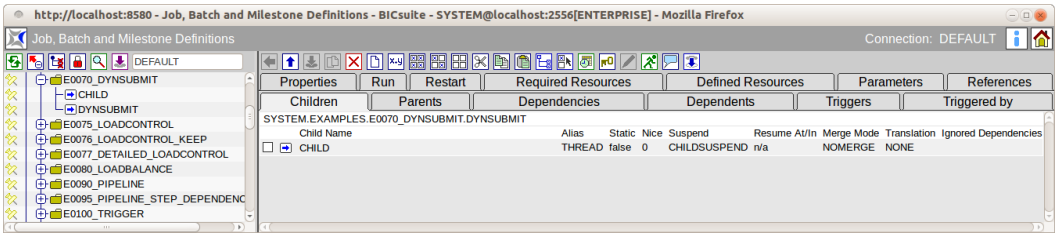


Abbildung 14.14: Batches und Jobs; Children Tab

Hier erscheint eine Liste von allen Job-, Milestone- und Batchobjekten, welche dem gewählten Job als Children untergeordnet sind. Die Spalten der obigen Liste haben folgende Bedeutung:

Child Name Das ist der Name des Childs. Durch Anklicken des Namens erscheint der Tab Child Details.
Die restlichen Spalten werden im nächsten Abschnitt erläutert.

14.5.4.1 Tab Child Details

Der Tab Child Details erscheint, wenn im Tab Children der Name eines Child-Objektes angewählt wurde. Hier werden alle Daten, welche schon in der Tabelle als Elemente angezeigt wurden, noch einmal gelistet und können geändert werden. Der Tab sieht folgendermaßen aus:

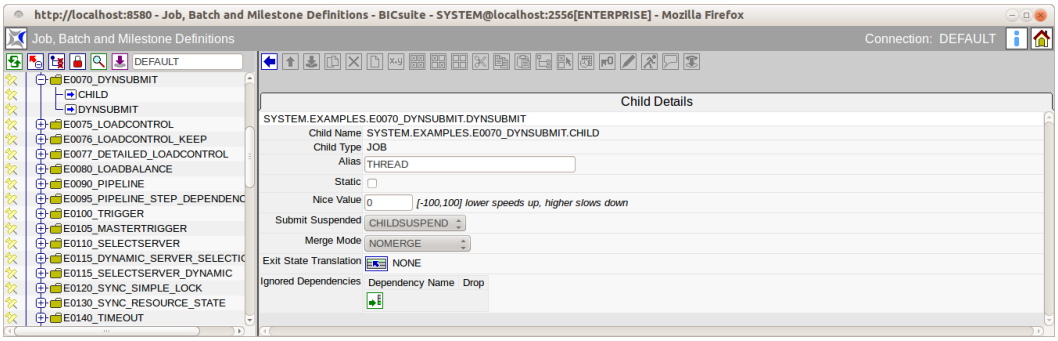


Abbildung 14.15: Batches und Jobs; Child Details

Die Felder des Tabs "Child Details" haben folgende Bedeutung:

Child Name Der *Child* Name wird immer mit dem gesamten Pfad der darüberliegenden Ordnerhierarchie angezeigt.

Child Type Der *Child Type* gibt die Art des Childs (Batch, Job oder Milestone) an.

Alias Mit dem *Alias* kann einem Child eine neue logische Benennung zugeordnet werden. Diese Benennung muss innerhalb aller Children des gewählten Jobs eindeutig sein. Der Alias ist nur bei Dynamic Children von Bedeutung. Der Parent Job kann beim Submitten durch Angabe des Alias weitere Instanzen der Children erzeugen. Damit ist im Parent kein Wissen über die Speicherstelle des Childs notwendig. Durch diese Abstraktion werden Umstrukturierungen der Folderstruktur erheblich vereinfacht.

Static Der Schalter *Static* gibt an ob ein Child statisch oder dynamisch submitted wird. Handelt es sich um ein statisches Child, so wird die Child-Instanz schon beim Submit des Parents instanziiert. Handelt es sich um ein dynamisches Child (der Static-Schalter ist nicht gesetzt), so können eine oder mehrere Instanzen des Childs zur Laufzeit dynamisch erzeugt werden.

Dynamische Children werden häufig bei Parallelisierungsaufgaben verwendet. Hier wird erst zur Laufzeit vom Run Program entschieden ob und wie viele dynamische Children angelegt werden sollen und diese werden durch die BICsuite-API angelegt. Hiermit ist eine beliebige Aufteilung eines Prozesses in viele parallel arbeitende Teilprozesse möglich.

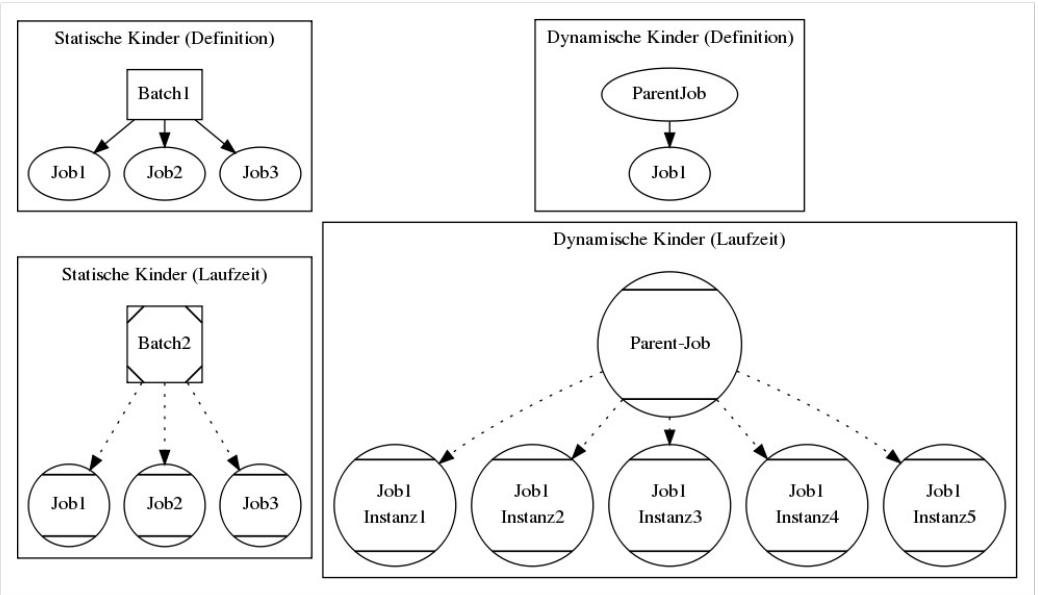


Abbildung 14.16: Statische und dynamische Children

Die Legende für die Grafik finden Sie im Kapitel 1.8.

Nice Value Der *Nice Value* gibt an, mit welcher Priorität das gewählte Child laufen soll. Der Nice Value gibt einen Offset an, der auf die Priorität des Childs addiert wird.
Beispiel: Hat der Prozess einen Nice Value von -50 und das Child hat eine Priorität von 100, dann hat die als Child von diesem Job gestartete Jobinstanz eine Priorität von 50. Wird der Child-Prozess alleine gestartet hat er eine Priorität von 100.
Handelt es sich beim Parent um einen Batch, wird der Nice Value zum Nice Value des Parents gezählt.

Submit Suspended Der Parameter *Submit Suspended* gibt an in welcher Form das Child Objekt beim Start verzögert wird oder ob er sofort gestartet werden kann. Es gibt folgende Optionen:

1. YES

Das Child wird beim Submit als Suspended angelegt und muss durch eine explizite Freigabe gestartet werden.

2. NO

   Das Child wird beim Submit nicht Suspended und kann sofort starten.

3. CHILDSUSPEND

   Ob eine Verzögerung stattfindet oder nicht, hängt vom Feld *Suspend* des Child-Jobs ab. Das heißt, je nachdem wie die Einstellung im Child-Job definiert wurde, wird diese übernommen.

Resume Wird nur angezeigt falls *Submit Suspended* auf "YES" gesetzt wurde.
Hier kann ausgewählt werden, ob ein automatischer Resume stattfinden soll.
Es gibt folgende Möglichkeiten:

- NO: wählt diese Funktionalität ab.
- AT: wählt einen automatischen Resume zu einem festen Zeitpunkt. Das Eingabefeld *Resume Time* wird angezeigt.
- IN: wählt einen automatischen Resume nach Ablauf einer Zeit. Die Eingabefelder *Resume In* und *Unit* werden angezeigt.

Resume Time Wird nur angezeigt falls im *Resume* "AT" gewählt wurde.
Hier wird der gewünschte Resume Zeitpunkt im Format "YYYY-MM-DDTHH:MI:SS" eingegeben.
Das Format orientiert sich an der ISO Norm 8601 und erlaubt auch unvollständige Angaben. Die Eingabe von 'T16:00' wird den Job um 16:00 Uhr resumen (ausgehend von der Zeit des Job Submits).

Resume In Wird nur angezeigt falls im *Resume* "IN" gewählt wurde.
Hier wird angegeben, wie viele Zeiteinheiten (siehe *Unit*) bis zum Resume gewartet werden soll.

Unit Wird nur angezeigt falls im *Resume* "IN" gewählt wurde.
Hier wird eingegeben, ob es sich bei der Eingabe im *Resume In* um Minuten (MINUTE), Stunden (HOUR), oder Tage (DAY) handeln soll.

Merge Mode Der Merge Mode gibt an, ob ein Child-Objekt mehrfach innerhalb eines "Master Job"-Laufes gestartet wird oder nicht.
Die Legende für die Grafik finden Sie im Kapitel 1.8.
Es gibt folgende Optionen:

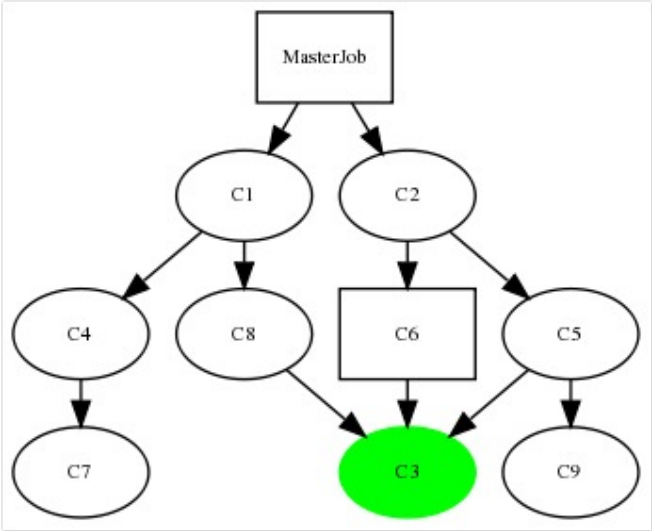


Abbildung 14.17: Beispiel für Merge Mode

1. No Merge

   Es erfolgt keine Zusammenfassung von Jobs. Der Child-Job wird gestartet ohne Berücksichtigung ob dieser Job im aktuellen Master Batch bereits gelaufen ist.

   Im Beispiel wie dargestellt im Bild 14.17 wird Job C3 dreimal gestartet, wenn in jeder Parent-Child-Beziehung ein No Merge steht.

2. Failure

   Zur Submit-Zeit des Jobs wird geprüft ob der Child-Job bereits gestartet wurde und falls dies der Fall ist, wird eine Fehlermeldung ausgegeben und der Start des Master Jobs wird abgebrochen.

   Im Beispiel wie dargestellt im Bild 14.17 kann der Job nicht submitted werden.

3. Merge Local

   In der Child-Hierarchie des Parent-Objektes wird ein Merge durchgeführt. Das heißt, wenn der Job in der Child-Hierarchie schon einmal gestartet worden ist, wird er nicht noch einmal gestartet. Wurde der Job innerhalb des Master Jobs, aber außerhalb der Child-Hierarchie des Parents schon einmal gestartet, so wird dies nicht berücksichtigt, sondern der Job neu submitted.

   Die Abhängigkeiten von diesem Job werden dabei genauso berücksichtigt, als ob dieser Job in dieser Child-Beziehung gestartet worden wäre.

Falls im oberen Beispiel, bei der Child-Beziehung C2→C6 und C2→C5 der Merge Mode Local gesetzt ist, wird der Job C3 nur zweimal gestartet, da in der Child-Hierarchie von C2 der Merge durchgeführt würde. Da die Beziehung C8→C3 nicht in dieser Child-Hierarchie liegt, wird dort C3 ausgeführt, da hier kein Merge durchgeführt wird.

4. Merge Global

   Es wird innerhalb des gesamten Master Jobs geprüft ob dieser Job schon einmal gestartet wurde und falls dies der Fall ist, wird der Job nicht neu gestartet.

   Dies kann nötig sein falls ein Job nur einmal laufen soll, aber von mehreren Jobs als Voraussetzung gebraucht wird. Er kann von mehreren Parent-Prozessen als Child definiert werden. Der Parent-Prozess der als erster ausgeführt wird, startet den Prozess. Die anderen Parent-Prozesse überprüfen durch das Merge Global ob dieser Job bereits gelaufen ist und starten ihn nicht noch einmal.

   Die Abhängigkeiten von diesem Job werden dabei genauso berücksichtigt, als ob dieser Job in dieser Child-Beziehung gestartet worden wäre.

   Im oberen Beispiel wird der Job C3 nur einmal gestartet, da ein globaler Merge alle Beziehungen im Master Job berücksichtigt und deshalb bei den Beziehungen (C8→C3, C6→C3 und C5→C3) den Merge durchführt.

   Durch welche Beziehung der Job nun wirklich gestartet wird, ist unerheblich.

Exit State Translation In dieser Auswahlmaske kann die Exit State Translation ausgewählt werden, die verwendet wird um dem Parent-Objekt den Exit State des Child-Objektes mitzuteilen. Mehr zu Exit State Translations finden Sie im Kapitel 5.

Liste Ignored Dependencies Hier kann eine Liste von Abhängigkeiten (Dependencies) hinzugefügt werden, welche innerhalb dieser Parent-Child-Beziehung vom Child ignoriert werden soll. All diese Abhängigkeiten werden nun für das Child nicht mehr berücksichtigt und das Child kann gestartet werden, ohne dass die Abhängigkeit des Parents erfüllt ist.
Ein Beispiel für Ignored Dependencies ist im Bild 14.18 abgebildet. Die Legende für die Grafik finden Sie im Kapitel 1.8.
Es können nur die Abhängigkeiten ignoriert werden, welche im Feld *Dependency Name* einen Wert eingetragen haben. Nicht benannte Abhängigkeiten können nicht in der Auswahlbox ausgewählt und dadurch nicht ignoriert werden.

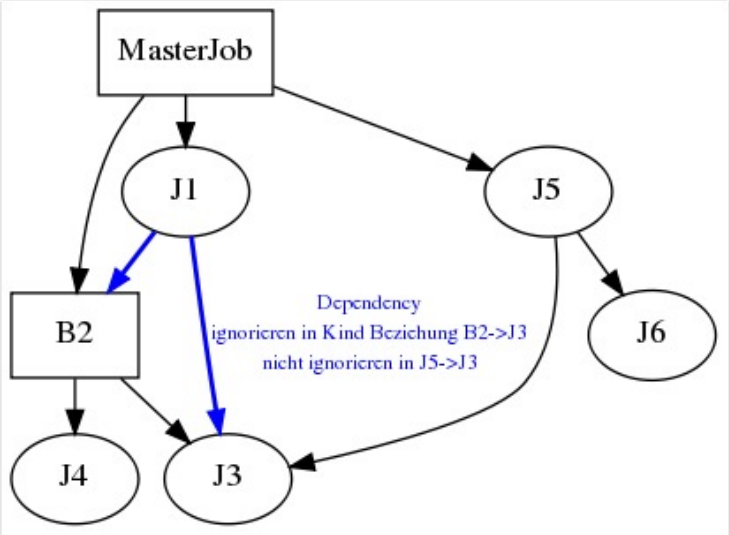


Abbildung 14.18: Beispiel Ignored Dependencies

14.5.5 Tab Parents

Der Tab Parents zeigt eine Liste von übergeordneten Parent-Objekten an, welche das gewählte Objekt als Child verwenden. Dies stellt also bildlich die Sicht nach oben dar, während der Children die Sicht nach unten darstellt, da dort alle Children des aktuell gewählten Objektes angezeigt werden. Der Tab sieht aus wie im Bild

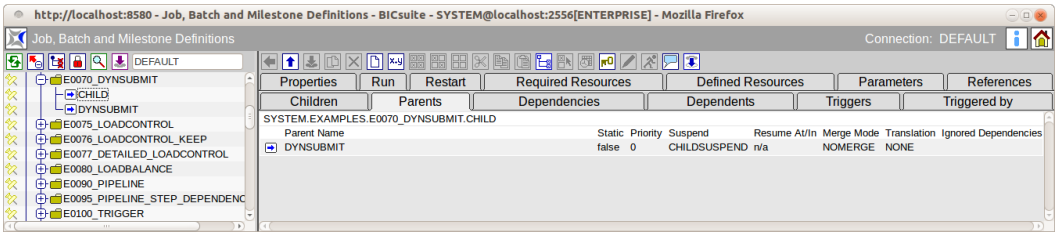


Abbildung 14.19: Batches und Jobs; Parents Tab

14.19.

Der Tab dient nur zur Information, es können keine Werte geändert oder neue Listeinträge hinzugefügt werden. Dies ist nur in der Parent-Child Richtung möglich.

Die Spalten der obigen Liste werden im Tab Children erläutert.

14.5.6 Tab Dependencies

Der Tab Dependencies zeigt eine Liste von allen Objekten an, von denen das aktuelle Objekt abhängig ist. Eine Abhängigkeit ist dann vorhanden, wenn das aktuelle Objekt nicht starten darf, bevor nicht alle Objekte von denen der Job abhängig ist vorher ausgeführt und mit einem gewissen Exit State beendet wurden.
Mittels Copy and Paste können weitere Objekte als benötigte Scheduling Entities (required Scheduling Entities) hinzugefügt werden.

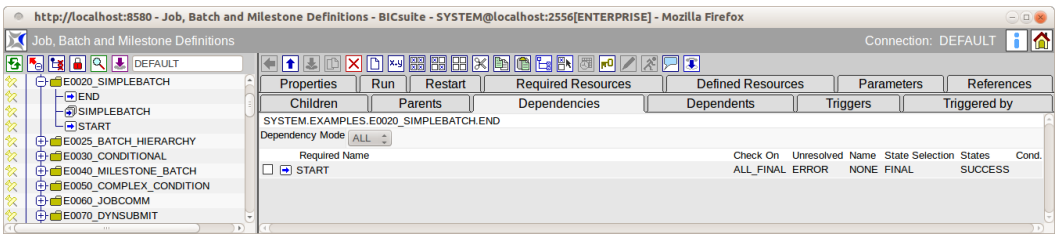


Abbildung 14.20: Batches und Jobs; Dependencies Tab

Der Tab sieht aus wie im Bild 14.20.
Das Feld der obigen Liste hat folgende Bedeutung:

Dependency Mode Der Dependency Mode gibt an, in welchem Zusammenhang die Liste der Dependencies gesehen werden muss. Es gibt folgende Optionen:

1. ALL

   Es müssen alle Abhängigkeiten gemeinsam erfüllt sein, damit das Objekt starten kann. Dies entspricht einer Verknüpfung aller Bedingungen mit **"UND"**.

   Beispiel: Objekt C ist von Job A und Job B abhängig (das heißt, er soll nur starten wenn A und B fertig sind) so muss der Dependency Mode ALL gewählt werden.

2. ANY

   Es muss mindestens eine der Abhängigkeiten erfüllt sein, damit das Objekt starten kann. Dies entspricht einer Verknüpfung aller Bedingungen mit einem **"ODER"**.

   Beispiel: Objekt C ist von Job A oder Job B abhängig (das heißt, er soll starten wenn A oder B fertig ist) so muss der Dependency Mode ANY gewählt werden.

Innerhalb der Dependencies eines Jobs kann es entweder die Verknüpfung mit ANY oder mit ALL geben. Sollen komplexere Abhängigkeitsverknüpfungen abgebildet werden (zum Beispiel abhängig von A oder B und C oder D) so wird

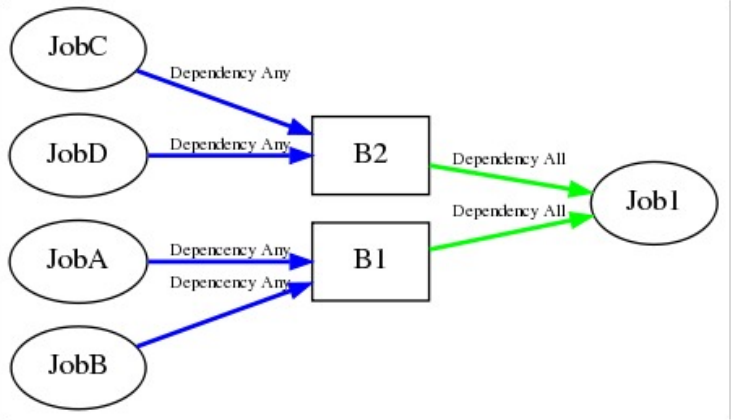


Abbildung 14.21: Beispiel für Dependency Modes

dies, wie im Bild 14.21 dargestellt, mittels der Verwendung eines Milestone- oder Batch-Objektes durchgeführt.
Die Legende für die Grafik finden Sie im Kapitel 1.8.
Im oben genannten Beispiel wurden Batch B1 und B2 generiert. B1 hat die Abhängigkeiten A und B mittels ANY verknüpft. B2 analog C und D ebenfalls mittels ANY. Anschließend kann unser Objekt (Job1) die beiden Batch mittels ALL als Abhängigkeit verknüpfen und hat die komplexe Bedingung (abhängig von A oder B und C oder D) hiermit abgebildet.
Die Liste "Dependencies" zeigt alle Objekte an, von denen das aktuell gewählte Objekt abhängig ist. Alle Elemente in der Liste werden mittels des Dependency Modes verknüpft.
Die Spalten der Liste werden im nächsten Abschnitt erläutert.

14.5.6.1 Tab Dependency Details

Der Tab Dependency Details erscheint, wenn im Tab Dependencies ein Eintrag durch Anklicken des Required Name ausgewählt wurde.
In den Dependency Details sind alle Aspekte einer Abhängigkeit zwischen den zwei Jobs einstellbar. Der Tab sieht folgendermaßen aus:
Die Felder des Tabs "Dependency Details" haben folgende Bedeutung:

Dependency Name Der *Dependency Name* ist optional und ist die Voraussetzung zum Ignorieren der Dependencies.

Required Name Hier steht der Name des Objektes, von dem der aktuell gewählte Job abhängig ist. Er entspricht dem vollen Namen, mit übergeordneter Ordnerhierarchie.

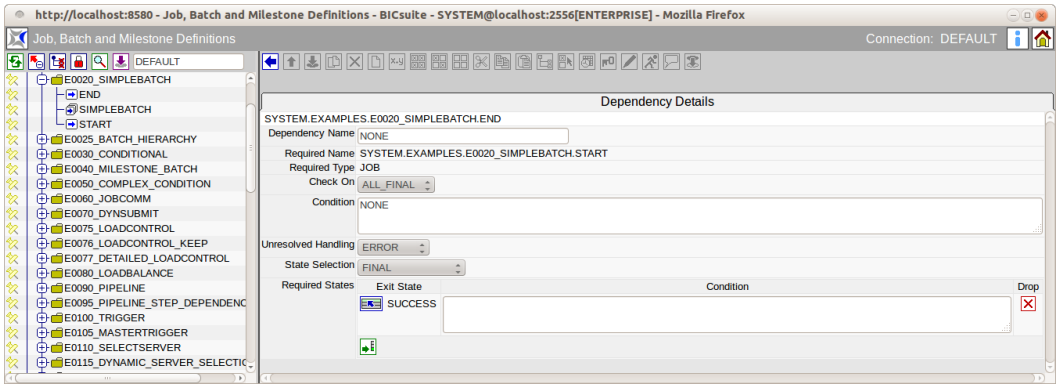


Abbildung 14.22: Batches und Jobs; Dependency Details

Required Type Im *Required Type* wird die Art des benötigten Objektes angezeigt.

Check On Im Feld *Check On* wird die Art der Vollständigkeitsprüfung für das benötigte Objekt beschrieben. Es gibt folgende Optionen:

1. ALL_FINAL

   Wird diese Option ausgewählt, so muss der benötigte Job einen Final State erreicht haben und alle seine Children ebenfalls.

2. JOB_FINAL

   Wird diese Option ausgewählt, so wird nur geprüft ob der Job einen Final State erreicht hat. Der State der Child-Objekte wird nicht geprüft.

Condition Die Condition ist eine Bedingung die zusätzlich erfüllt sein muss damit die Abhängigkeit als erfüllt gilt. Parameter vom Typ "Resourcereference" können nicht benutzt werden, denn zum Zeitpunkt der Auswertung der Condition sind noch keine Ressourcen belegt.

Unresolved Handling Im Auswahlfeld *Unresolved Handling* wird beschrieben was zu tun ist, wenn eine abhängige Objektinstanz im aktuellen Master Batch nicht vorhanden ist.
Die Legende für die Grafik finden Sie im Kapitel 1.8.
Es gibt folgende Optionen:

1. IGNORE

   Die Abhängigkeit wird ignoriert, falls der benötigte Job nicht vorhanden ist. Ist der Job vorhanden, wird sie ganz normal beachtet.

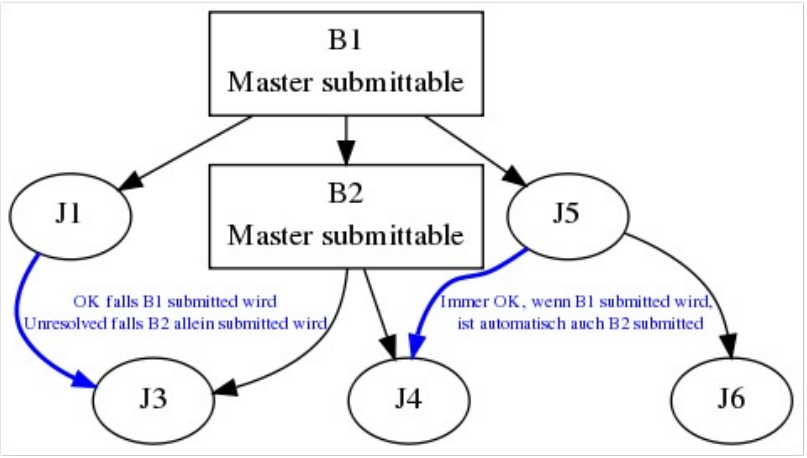


Abbildung 14.23: Beispiel für Unresolved Handling

Ein Ignore kann dann nötig sein, wenn kleinere Batches zu größeren zusammengefasst werden. Läuft der kleine Batch alleine, so kann er alle Abhängigkeiten, die nur beim größeren Lauf zu beachten sind, ignorieren. Falls er als Teil des großen Batches läuft, sind die benötigten Jobs dann vorhanden und die Abhängigkeiten müssen beachtet werden.

Im oberen Beispiel muss in der Abhängigkeit J1→J3 ein Ignore eingetragen werden, wenn der Batch B2 auch allein gestartet werden darf.

2. ERROR

   Ist der benötigte Job nicht vorhanden, so wird der Submit-Vorgang mit einem Fehler beendet.

3. SUSPEND

   Der aktuelle Job geht in den Suspend State. Das heißt, er muss durch ein externes Programm oder manuellen Eingriff gestartet werden, sobald der benötigte Job vorhanden ist.

   Im oberen Beispiel würde Job J3 in den Suspend State gehen, falls J1 ein dynamisches Child von B1 wäre und zeitlich nicht klar ist, wann der Job submitted wird.

   Die Idee hinter ”Suspend” ist, dass weder ein unbewachtes Starten, noch ein ”ERROR” auftritt.

State Selection Im Feld *State Selection* definiert welche Exit States des benötigten Jobs die Abhängigkeit erfüllen und der abhängige Job starten kann. Es gibt folgende Optionen:

1. FINAL

   Wird diese Option ausgewählt, so muss der benötigte Job einen Final State erreicht haben. Diese Option ermöglicht die Auswahl der benötigten Exit States in der nachfolgenden Tabelle *Required States*.

2. ALL_REACHABLE

   Wird diese Option ausgewählt, sind alle Final Exit States gültig welche im Exit State Profile des benötigten Jobs nicht als Unreachable gekennzeichnet wurden.

3. UNREACHABLE

   Wird diese Option ausgewählt, ist nur der Exit State gültig welcher im Exit State Profile des benötigten Jobs als Unreachable gekennzeichnet wurde.

4. DEFAULT

   Wird diese Option ausgewählt, sind alle Final Exit States gültig welche im Exit State Profile des benötigten Jobs als Dependency Default gekennzeichnet wurden.

Required States Hier steht die Liste von allen gültigen Exit States, welche das benötigte Objekt haben muss, damit die Abhängigkeit erfüllt und der abhängige Job starten kann. Diese Liste ist nur sichtbar wenn im Feld *State Selection* die Option FINAL gewählt wurde.
Beispiel:
Ein vom Job B benötigter Job A kann zwei Exit States (SUCCESS oder WARNING) annehmen. Da der Job B nur starten soll, wenn Job A mit einen Exit State SUCCESS beendet wurde (also wenn er erfolgreich durchgelaufen ist), darf in der Liste der Required States nur ein Eintrag mit SUCCESS stehen. Ist Job A durchgelaufen und hat den Exit State SUCCESS erhalten, so kann Job B starten. Beendet sich Job A mit WARNING, kann Job B nicht starten.
Die Liste ist notwendig, da es möglich ist dass ein abhängiger Job mehreren Exit States als Voraussetzung akzeptiert. In unserem Beispiel heißt das, Job B hätte beide Exit States (SUCCESS und WARNING) als Required States eingetragen wenn er auf jeden Fall starten soll, egal ob Job A vollständig erfolgreich war oder mit einer Warnung beendet wurde.

14.5.7 Tab Dependents

Der Tab Dependents zeigt alle Scheduling Entities an, welche von dem gewählten Scheduling Entity abhängig (dependent) sind. Dies ist die umgekehrte Sicht wie in Tab Dependencies, da jetzt die abhängigen (dependent) Jobs angezeigt werden und nicht mehr die benötigten (required) Jobs.

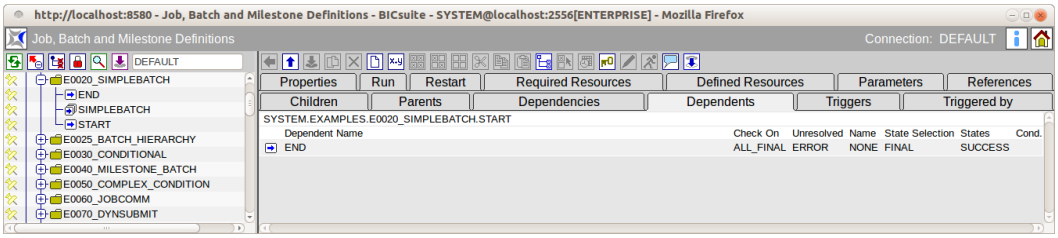


Abbildung 14.24: Batches und Jobs; Dependents Tab

Der Tab sieht folgendermaßen aus:
Die Liste zeigt alle Objekte (Jobs, Milestones, Batches), welche das gewählte Scheduling Entity benötigen, an. Die Liste ist rein informativ und kann weder geändert, noch können die Feldwerte in dieser Ansicht geändert werden. Um eine Änderung durchzuführen, muss in das in der Liste angezeigte Objekt navigiert und dort der Tab Dependencies aufgerufen werden. Hier ist die Änderung möglich.
Die Spalten der Liste werden im Tab "Dependency Details" erläutert.

14.5.8 Tab Required Resources

Der Tab Resources zeigt alle, für die Ausführung dieses Objektes, notwendigen Resources an. Weitere notwendige Resources können hier hinzugefügt werden.
Es werden alle, bereits durch das Environment oder einen angegebenen Footprint definierten, Ressourcen angezeigt. Diese können nur durch Änderung des entsprechenden Environments bzw. Footprints entfernt werden.
Der Tab Resources sieht folgendermaßen aus:

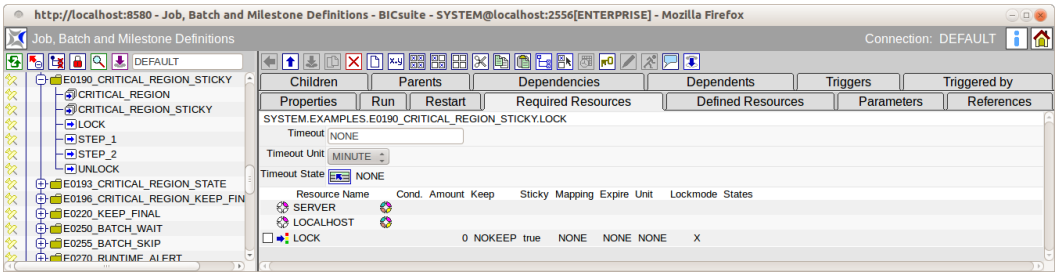


Abbildung 14.25: Jobs; Required Resources Tab

Die Felder und Spalten des Tabs "Required Resources" haben folgende Bedeutung:

Timeout Der *Timeout* gibt an wie lange nach dem Submit des Jobs auf alle Resources gewartet werden soll. Ist hier kein Wert angegeben, so wird unendlich lange auf die Resources gewartet.

Der Wert in diesem Feld muss im Zusammenhang mit dem Feld *Timeout Unit* gesehen werden. Im Feld *Timeout* wird der Wert des Timeouts eingegeben. Im Feld *Timeout Unit*, die Einheit.
Beispiel: Steht im Feld *Timeout* der Wert 5 und die Timeout Unit ist **MINUTES**, so wartet ein Job 5 Minuten auf eine Resource und falls diese in dieser Zeit nicht verfügbar wird, kann dieser Job nicht gestartet werden und beendet sich mit dem Exit State, welcher im Feld *Timeout State* definiert wurde.

Timeout Unit Die Timeout Unit gibt die Einheit an, welche für den gewählten Wert aus dem Feld *Timeout* anzuwenden ist. Ist im Feld *Timeout* kein Wert eingetragen, so spielt der Wert des Feldes *Timeout Unit* keine Rolle.

Timeout State In diesem Auswahlfeld kann gewählt werden, welche Exit State der Job annehmen soll, falls ein Timeout eingetroffen ist. Wird der Auswahlknopf gedrückt, so kann aus der Liste die Exit States ausgewählt werden, die vom *Exit State Profile* zur Verfügung gestellt wurden.
Es könnte hier zum Beispiel ein eigener Timeout State definiert werden oder es wird der Failure State verwendet.
Ist im Feld *Timeout* kein Wert eingetragen, so spielt der Wert des Feldes *Timeout State* keine Rolle.
In der Liste "Resources" werden alle Ressourcen angezeigt die aktuell vom gewählten Job benötigt werden um ausgeführt werden zu können. Die Liste enthält folgende Spalten:

Resource Name Das ist der Name der Resource, welche für den Start des aktuellen Jobs benötigt wird.
Nach dem Namen des Feldes ist ein Icon angegeben, welches angibt, aus welchem Zusammenhang die Resource stammt. Folgende Icons werden angezeigt:

- Footprint

  Die Resource stammt aus dem Feld *Footprint*. Für jede Resource kann eine individuelle Anforderung angelegt werden. Diese überschreibt die ursprüngliche Anforderung.

- Environment

  Die Resource stammt aus dem Feld *Environment*. Er ist nur aus Informationszwecken vorhanden und kann in diesem Tab nicht geändert werden.

- Kein Icon

  Ist nach dem Namen der Resource kein Icon vorhanden, so handelt es sich um eine in diesem Tab hinzugefügte Resource, welche auch hier änderbar ist. Durch das Anklicken des Namens kann in den Tab Resource Details gesprungen und die Werte in der Liste geändert werden. Darüber hinaus ist es möglich, durch Anklicken des *Selektions*-Buttons vor dem Namen, diese Zeile zu markieren und zu löschen oder zu verschieben.

Cond. Im Feld *Cond.* kann eine Bedingung stehen, die erfüllt sein muss, damit die Resource als gültig anerkannt wird. Die restlichen Spalten werden im nächsten Abschnitt erläutert.

14.5.8.1 Tab Resource Details

Der Tab Resource Detail erscheint, wenn im Tab Required Resources ein Listeneintrag angewählt wurde.
Im Tab Resource Detail können alle Detailinformationen zu einer zugeordneten Resource eingetragen und geändert werden.
Der Tab sieht folgendermaßen aus:

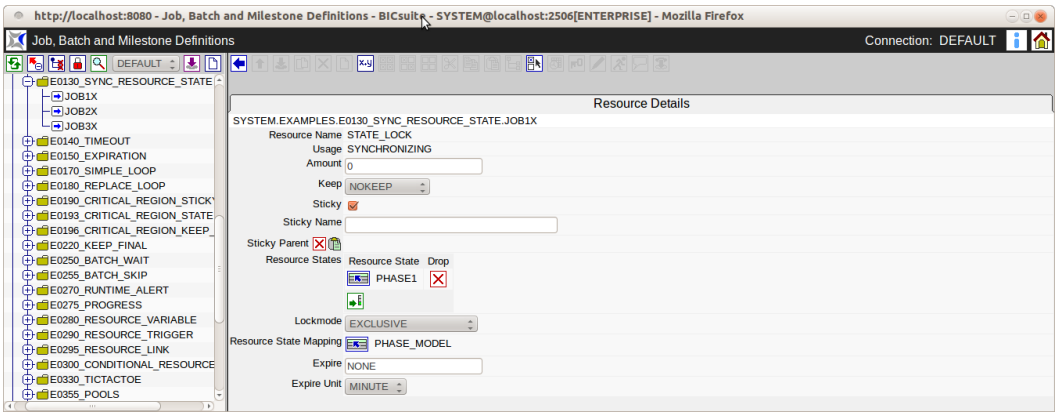


Abbildung 14.26: Batches und Jobs; Resource Requirement Details

Die Felder des Tabs ”Resource Details” haben folgende Bedeutung:

Resource Name Hierbei handelt es sich um den Namen der benötigten Resource. Es wird jeweils der volle Name mit der darüberliegenden Ordnerhierarchie angezeigt.

Usage Hierbei handelt es sich um die Usage der Resource. Weitere Informationen zur Usage finden Sie im Kapitel 10.3.3.

Amount Dieses Feld wird nur angezeigt, falls die Usage vom Typ "SYSTEM" oder "SYNCHRONIZING" ist.
Das ist die Menge die von der Resource in diesem Job benötigt wird. Im Jobserver (oder dem darüberliegenden Scopes, siehe Jobserver) werden maximale Mengen der Resource zur Verfügung gestellt. Jeder Job, der von dieser Resource eine Menge benötigt, reduziert die Anzahl der freien Mengen.
Beispiel:
Ein Jobserver stellt von der Resource A die Menge 5 zur Verfügung (zum Beispiel 5 CPU Einheiten auf diesem Server).
Job A startet nun und benötigt eine Menge von 3. Der Jobserver hat nun 3 CPU Einheiten belegt und 2 noch verfügbar.
Job B möchte starten, benötigt aber ebenfalls einen Amount von 3. Der Job kann nicht an den Jobserver zum Starten übergeben werden, da dieser im Moment nur noch 2 CPU Einheiten zur Verfügung hat.
Job B kann erst starten, wenn Job A beendet ist und mindestens drei CPU Einheiten zur Verfügung stehen.

Keep Hier wird der Wert des Keep Parameters eingestellt. Weitere Informationen zum Keep Parameter finden Sie im Kapitel 12.3.1.

Sticky Mittels dieses Schalters kann die Resource-Rückgabe gesteuert werden. Ist dieser Schalter gesetzt, so wird die Resource innerhalb eines Master Jobs solange behalten, bis der letzte Job, der diese Resource mit eingeschalteten Sticky Flag benötigt, beendet wurde.
Dies dient dazu, um einmal reservierte Ressourcen innerhalb eines Master Jobs zu behalten, ohne das die Resource von anderen Jobs in der Zwischenzeit verwendet werden darf.
Als Beispiel:
Master Job M1 besteht aus den Jobs A, B, C und D, die nacheinander ausgeführt werden. Job A generiert eine Tabelle-X, welche von Job D benötigt wird. Die Jobs B und C benötigen diese Tabelle-X nicht. Andere Master Jobs (M2) möchten auch auf die Tabelle zugreifen und diese verändern. Solange Job D aber noch nicht gelaufen ist, soll dies verhindert werden, da diese Aktionen das Ergebnis des Jobs A verfälschen und unbrauchbar machen würden. Die Tabelle-X wird nun als Synchronizing Resource abgebildet.
Ohne gesetztes Sticky Flag, würde die Resource nach dem Ende des Jobs A zurückgegeben werden und die Tabelle könnte von J2 verändert werden. Mit gesetztem Sticky Flag (welches in Job A und in Job D gesetzt werden muss) wird die Resource nach der Beendigung des Jobs A nicht zurückgegeben, sondern im Master Job als belegt gehalten.
Andere Jobs können diese nun nicht ändern. Erst nach der Beendigung des Jobs D (des letzten Jobs in diesem Master Job, der die Resource mit gesetztem Sticky Flag benötigt), wird die Resource freigegeben und andere Jobs können diese verwenden.

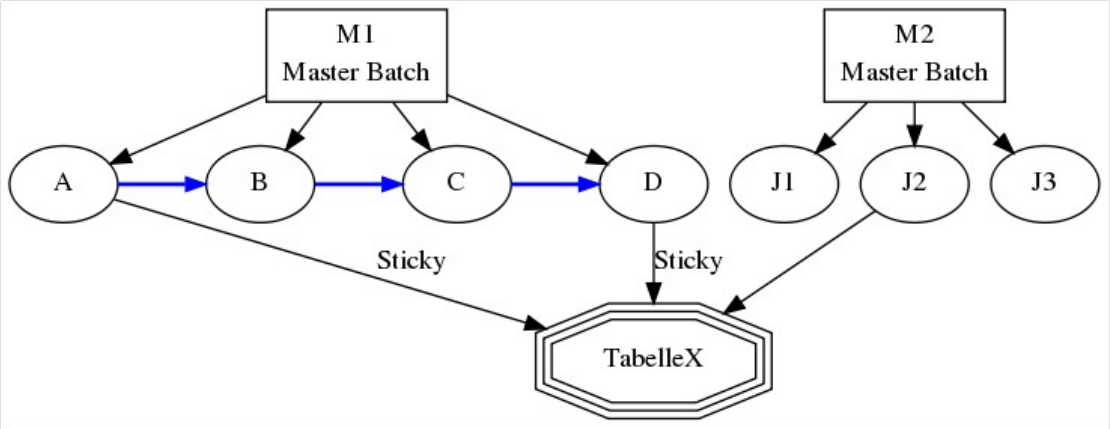


Abbildung 14.27: Beispiel für Sticky Handling

Mit dem Sticky Flag kann man über eine Sticky Resource alle notwendigen Prozesse auf dem selben Scope bzw. Jobserver ausführen lassen.
Hier ein weiteres Beispiel zur Erläuterung:

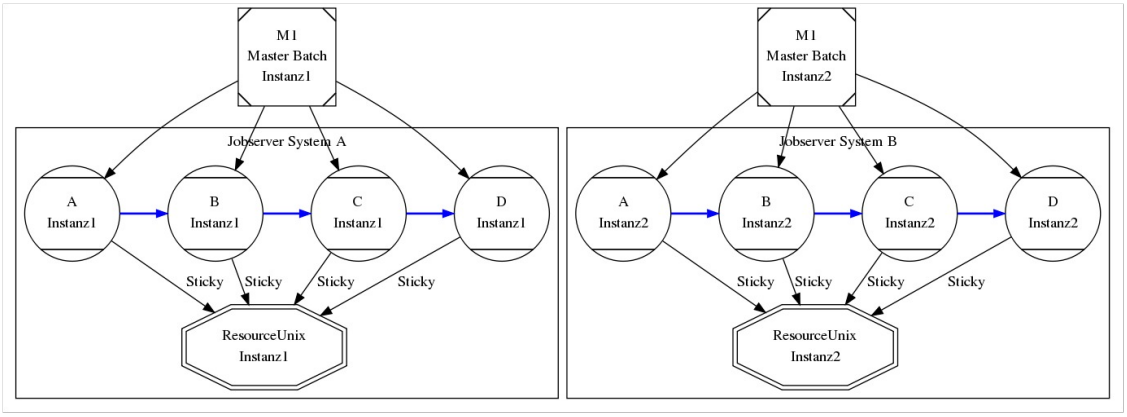


Abbildung 14.28: Beispiel für Lastverteilung mit Sticky Handling

Es gibt in einer Systemumgebung 2 Unix-Rechner mit identischer Konfiguration und Umgebung, die zur Lastverteilung eingesetzt werden. Nun kann ein Master Job entweder auf der einen oder auf der anderen Maschine laufen. Da aber Zwischenergebnisse als Files zwischen den einzelnen Jobs des Master Jobs übertragen werden sollen, muss der gesamte Master Job, nachdem die Maschine ausgewählt wurde, zwingend auf dieser laufen.

Dies kann mittels des Sticky Flags erreicht werden. Es werden auf diesen Maschinen jeweils Jobserver (Jobserver System A und System B) installiert und die Umgebung wird als eine Synchronizing Resource Definition (Resource Unix) abgebildet. Innerhalb der beiden Jobserver wird nun diese Resource Definition zugeordnet (2 mal, in jeden Jobserver 1 mal). Nun müssen alle Children des Master Jobs so definiert werden, dass sie die Resource Definition mit dem Sticky Flag benötigen.
Der erste Job (Job A) des Master Jobs belegt nun eine der beiden verfügbaren Resources (auf dem einen oder dem anderen Jobserver). Da diese Sticky belegt wurde, wird sie nach Beendigung des ersten Jobs nicht mehr zurückgegeben. Alle weiteren Jobs des Master Jobs sind über diese Sticky Resource auf den gewählten Jobserver und damit auf diese Maschine festgelegt.
In der Grafik läuft nun ein Master Job (Instanz 1) vollständig auf dem Jobserver System A und ein zweiter Master Job (Instanz 2) vollständig auf dem Jobserver System B.

Sticky Name Ist das oben beschriebene Sticky Flag gesetzt, so wird auch das Feld Sticky Name sichtbar. Ist der Sticky Name gesetzt, so steht die Ressource nicht allen Jobs des selben Master Runs zur Verfügung, sondern nur den Jobs, welche die Resource mit dem selben Sticky Name allokieren wollen.

Sticky Parent Ist das oben beschriebene Sticky Flag gesetzt, so wird auch das Feld Sticky Parent sichtbar. Ist der Sticky Parent gesetzt, so steht die Ressource nicht allen Jobs des selben Master Runs zur Verfügung, sondern nur den Jobs, welche die Resource mit dem selben Sticky Parent allokieren wollen.

Liste Resource States Die Liste Resource States wird nur angezeigt, wenn die Resource vom Typ ”Synchronizing” ist und ein Resource State Profile zugeordnet wurde.
Hier kann eine Liste von benötigten Resource States eingetragen werden. Der Job kann nur dann starten, wenn die benötigte Resource in einem dieser States ist.

Lockmode Der Lockmode gibt an, mit welchem Zugriffsmodus auf die benötigte Resource zugegriffen werden soll. Weitere Informationen zum Lockmode finden Sie im Kapitel 13.4.2.3.

Resource State Mapping Dieses Feld wird nur angezeigt, wenn die Resource vom Typ ”Synchronizing” ist und ein Resource State Profile zugeordnet wurde.
Das Resource State Mapping gibt an, wie und ob der State der Resource nach Beendigung des Jobs geändert werden soll. Weitere Informationen zum Resource State Mapping finden Sie im Kapitel 8.

Expire Gibt den Wert des Expiration-Intervalls an. Weitere Informationen zu Expire finden Sie im Kapitel 10.3.6.

Expire Unit Gibt die Einheit des Expiration-Intervalls an. Weitere Informationen zu Expire Unit finden Sie im Kapitel 10.3.6.

14.5.9 Tab Defined Resources

Die Defined Resources werden im dem Moment instanziiert in dem der Job startet. Diese Resources sind nur für Jobs sichtbar die im gleichen Ablauf sind.

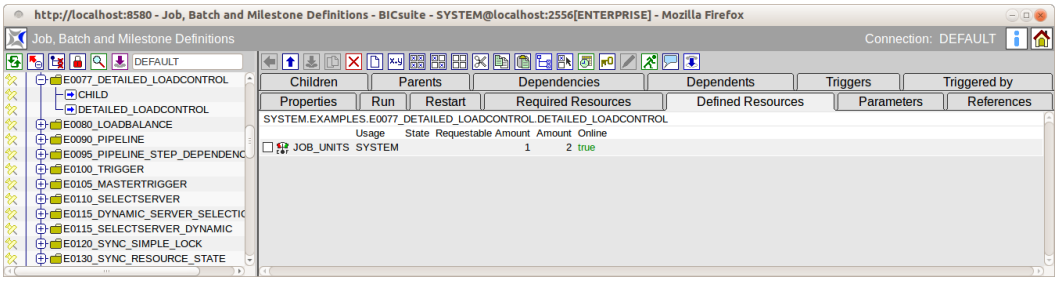


Abbildung 14.29: Batches und Jobs; Defined Resources

Die Felder der obigen Liste haben folgende Bedeutung:

Usage Die *Usage* gibt an, um welchen Typ "Resource" es sich handelt. Mehr zu Typen von Ressourcen finden Sie im Dialog Named Resource.

State Das Feld *State* ist nur zu sehen, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde. Hierbei handelt es sich um den aktuellen Status der Resource in diesem Scope oder Jobserver. Der Status kann mittels dieses Wertes gesetzt bzw. geändert werden. In der "Drop Down" Liste sind alle gültigen Resource States des Resource State Profiles enthalten und können ausgewählt werden.
Dies kann bei einer manuellen Fehlerbehebung notwendig sein, um eine Resource wieder in einen Status zu bringen, in dem eine Weiterverarbeitung durch Folgejobs möglich ist.
Falls hier der Wert manuell geändert wird, wird der Wert des Feldes *State* im Tab Resources nicht automatisch aktualisiert. Falls dies gewünscht und notwendig ist, kann dies mittels des *Refresh* Buttons durchgeführt werden.

Requestable Amount Die Menge der Ressourcen die maximal von einem Job angefordert werden darf.

Amount Beim *Amount* handelt es sich um die Anzahl der vom aktuellen Job belegten oder angeforderten Resource-Instanzen. Übersteigt der Wert des Amounts die aktuell verfügbare Anzahl, die im Feld *Amount* im Tab Resource Detail gepflegt

werden können und ist kein alternativer Scope mit einer ausreichenden Anzahl vorhanden, kann der Job nicht ausgeführt werden.

Online Mittels *Online* ist es möglich eine Resource in diesem Scope oder Jobserver on- bzw. offline zu schalten. Ist die Resource offline, so steht diese Resource vorübergehend nicht zur Verfügung.

14.5.10 Tab Parameters

Im Tab Parameters können Parameter definiert werden, welche zur Kommunikation und Datenübermittlung zwischen Jobs verwendet werden können. Auch Parameter, welche von der Umgebung oder aus dem Scope bzw. Ordnerumfeld der Jobserver und Jobumgebung kommen, können hier für die Verwendung innerhalb des aktuellen Jobs definiert werden.
Der Tab Parameters sieht folgendermaßen aus:

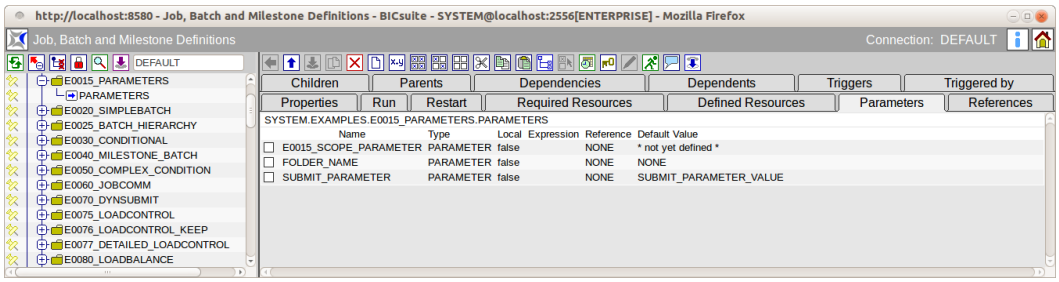


Abbildung 14.30: Batches und Jobs; Parameters Tab

In der obigen Liste sind alle Parameter aufgelistet, welche aktuell in diesem Scheduling Entity definiert wurden. Mittels des *New* Buttons kann ein neuer Parameter angelegt werden. Nach Eingabe der Daten und Rückkehr in die Liste, erscheint dieser auch hier. Mittels des *Löschen*-Buttons können ausgewählte Parameter auch gelöscht werden. Durch Anklicken des Parameternamens kommt man in den Tab "Parameter Details".
Die Spalten des Tabs "Parameters" werden in nächsten Abschnitt erläutert.

14.5.10.1 Tab Parameter Details

Der Tab Parameter Details erscheint, falls im Tab Parameters ein Parameter ausgewählt wurde oder mittels des *New* Buttons ein neuer Parameter angelegt werden soll. Hier können alle Detailinformationen eines Parameters eingetragen und geändert werden.
Der Tab Parameter Details sieht folgendermaßen aus:
Die Felder des Tabs "Parameter Details" haben folgende Bedeutung:

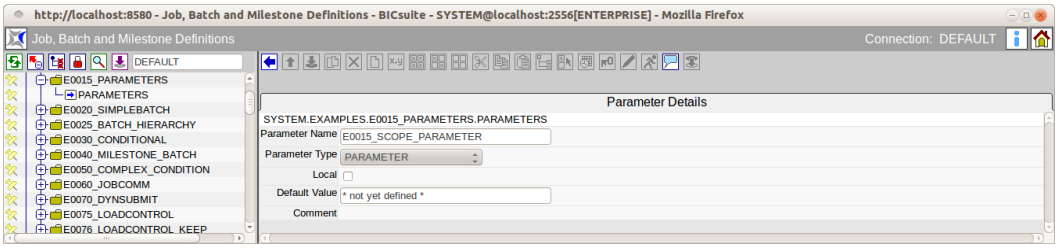


Abbildung 14.31: Batches und Jobs; Parameter Details

Parameter Name Hierbei handelt es sich um den Namen des Parameters. Durch den Namen erfolgt die komplette Werteübergabe der Parameter zwischen einzelnen Jobs. Der Wert des Parameters wird vom ersten Job gesetzt und kann im zweiten über den identischen Namen verwendet werden.

Parameter Type Beim Type handelt es sich um die Art des Parameters und wie er verwendet wird.
Optionen für Parameter Type

1. PARAMETER

   Parameter vom Typ ”Parameter” können zum Zeitpunkt des Submits spezifiziert werden, wenn ein Default-Wert spezifiziert wurde oder müssen, wenn kein Default spezifiziert wurde. Dies gilt allerdings nur für die Parameter des Master Batches. Alle anderen Parameter verhalten sich als wären sie vom Typ ”Import”.

2. IMPORT

   Der Wert muss erst zur Ausführungszeit auflösbar sein. Die Verwendung der Variable wird explizit dokumentiert.

3. RESULT

   Das Result stellt das Ergebnis eines Jobs dar, welches an einem Nachfolgejob exportiert werden kann.

4. CONSTANT

   Der Parameter ist eine Konstante die nicht mehr verändert werden kann. Der Wert der Konstanten wird mittels des Feldes *Default Value* eingetragen. Dieser Wert ist im Falle des Typs ”Constant” zwingend erforderlich.

5. EXPRESSION

   Ein Expression Parameter stellt eine Sonderform eines Parameters für parallele Verarbeitung mit dynamischen Child-Jobs dar. Um in diesem Fall ein Aggregat über alle Children zu bekommen, kann im zusätzlich erscheinenden

Feld ”Expression” eine Aggregatsfunktion ausgewählt und im Feld ”Expression Parameter Name” den zu aggregierenden Child-Parameter angegeben werden.

Wird der Expression Parameter zur Laufzeit benutzt, wird in allen Instanzen der Child-Jobs nach dem Wert dieses Parameters gesucht und die dementsprechende Aggregatsfunktion ausgeführt.

Ein Beispiel zur Verdeutlichung:

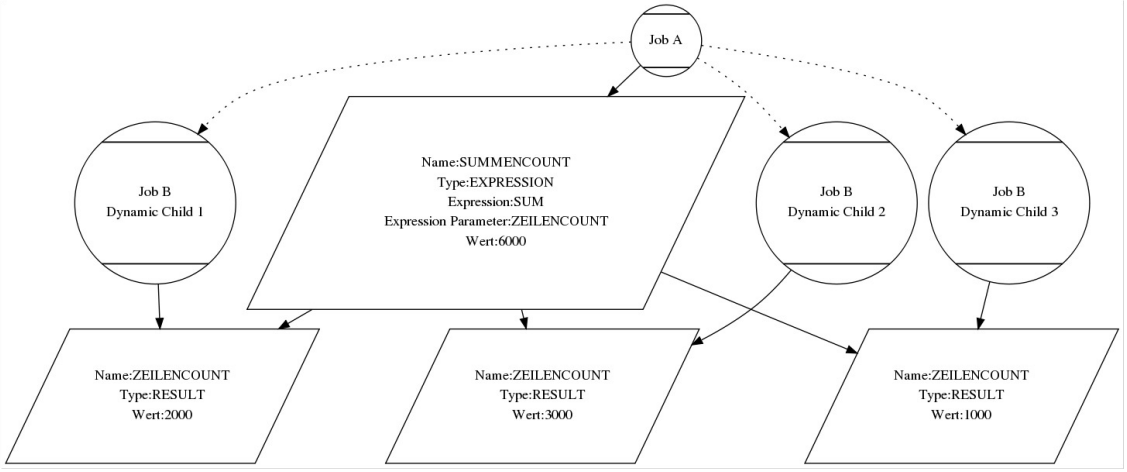


Abbildung 14.32: Beispiel für Expression

Job A hat einen dynamischen Child-Job B, welcher zur Laufzeit parallel submitted wird. Die Child-Jobs B schreiben alle x-Zeilen einer Tabelle parallel. Nun möchte A gerne wissen, wie viele Zeilen alle Child-Jobs in Summe geschrieben haben.

Um dies zu erreichen, muss Job B einen Parameter (Type Result) ZEILENCOUNT definiert haben, der beim Abschluss des Joblaufes die Anzahl der bearbeiteten Zeilen enthält.

In Job A wird nun ein Parameter SUMMENCOUNT angelegt, mit dem Typ ”Expression”, mit der Expression-Option ”SUM” und dem Parametername ”ZEILENCOUNT”.

Nun startet Job A und submitted 3 Child-Jobs B (B1, B2, B3). B1 schreibt 2000 Zeilen, B2 schreibt 3000 Zeilen, B3 schreibt 1000 Zeilen. Nach Beendigung der Children fragt Job A (oder ein Nachfolge-Job) den Parameter SUMMENCOUNT ab und erhält als Ergebnis: 6000 Zeilen, was der Summe aller Children entspricht.

6. REFERENCE

Durch den Typ "Reference" kann ein Parameter als Referenz eines anderen Parameters vergeben werden. Dieser Parameter wird durch eine Quersuche in allen Jobs, welche mit dem aktuellen Job auf einer Ebene liegen, durchgeführt. Hiermit ist es möglich Parameter von Jobs zu bekommen, welche nicht unmittelbar ein Parent des aktuellen Jobs sind. Der Parameter der referenziert werden soll muss im Feld *Reference* mittels *Paste* eingefügt werden (*Copy* aus dem Parameter Tab einer anderen Job Definition).

7. CHILDREFERENCE

   Durch den Typ "Childreference" kann ein Parameter als Referenz eines anderen Parameters vergeben werden. Dieser referenzierte Parameter wird durch eine Suche in allen Children des aktuellen Jobs ermittelt. Hiermit ist es möglich Parameter von den eigenen Children zu ermitteln.

8. RESOURCEREFERENCE

   Durch den Typ "Resourcereference" kann ein Parameter als Referenz eines Resource Parameters vergeben werden. Der Parameter der referenziert werden soll, muss im Feld *Reference* mittels *Paste* eingefügt werden (*Copy* aus dem Parameter Tab einer Named Resource).

Expression Dieses Feld erscheint nur falls im Feld *Parameter Type* der Wert "Expression" gewählt wurde.
Hier kann die Art der Aggregatsfunktion ausgewählt werden.
Bei allen Aggregatsfunktionen gilt: Sollte in manchen Child-Objekten der Parameter oder der Parameterwert nicht vorhanden oder vom falschen Typ sein (String statt Zahl), dann werden diese Werte ignoriert und nur die zur Verfügung stehenden und korrekten Werte aggregiert.
Es stehen folgende Aggregate zur Verfügung:

1. SUM

   Die Summe aller Werte des referenzierten Parameters aller Child-Elemente.

2. AVG

   Der Durchschnitt aller Werte des referenzierten Parameters aller Child-Elemente, welche gültige und vorhandene Werte enthalten.

3. MIN

   Das Minimum aller Werte des referenzierten Parameters.

4. MAX

   Das Maximum aller Werte des referenzierten Parameters.

5. COUNT

   Die Anzahl der Child-Instanzen, welche diesen Parameter definiert haben.

6. NONE

   Es wurde noch keine Aggregatsfunktion gewählt. None stellt keinen gültigen Wert dar.

Expression Parameter Name Dieses Feld erscheint nur, falls im Feld *Parameter Type* der Wert "Expression" gewählt wurde.
Das ist der Name des Child-Parameters, auf dem das Aggregat gebildet werden soll. Weitere Informationen zu Aggregaten finden Sie im Feld *Expression*.

Local Wenn das Häkchen gesetzt ist, ist dieser Parameter nur vom Job selbst sichtbar.

Default Value Der *Default Value* gibt an, welcher Wert der Standardwert ist. Dieser Wert wird bei der Parameterauflösung zurückgegeben, wenn bei der Suche kein explizit gesetzter Wert gefunden wird. Das Feld ist optional. Im Falle einer Konstante muss das Feld einen Wert besitzen.

Variablen werden rekursiv ausgewertet. Das bedeutet, wenn innerhalb einer Variable etwas der Form $NAME bzw. $NAME auftritt, wird die Variable NAME aufgelöst und der Wert an der Stelle eingesetzt. Auf diese Art und Weise ist es nun auch möglich geworden - indirekt - auf Systemvariablen anderer Jobs zuzugreifen. Ein $-Zeichen wird durch \$ dargestellt. Ein Backslash durch \\.

Comment Das Feld *Comment* dient als Möglichkeit einer näheren Erläuterung (Kommentar) des Objektes.

Reference Dieses Feld erscheint nur, falls im Feld *Parameter Type* der Wert "Reference", "Childreference" oder "Resourcereference"gewählt wurde.
Hier kann mittels *Copy and Paste* ein Parameter eingetragen werden, welcher über den gewählten Namen referenziert wird.

14.5.10.2 Predefined Standardparameter des Laufzeitsystems

Predefined Standardparameter im Laufzeitsystem können von allen Submitted Entities im Feld *Run Program* als Übergabeparameter oder im Run Program selbst verwendet werden. Die Predefined Standardparameter enthalten Informationen über das Submitted Entity und dessen Umgebung.
Es werden folgende Standardparameter vom Laufzeitsystem angeboten:

Job-Umgebungsparameter Hier handelt es sich um alle Parameter, welche mit dem Submitted Entity selber zu tun haben. Job Umgebungsparameter-Namen:

1. JOBNAME

   Das ist der Name des Jobs wie er im Feld *Name* angegeben wurde.

2. JOBID

   Hierbei handelt es sich um die ID des Submitted Entities. Hiermit ist eine eindeutige Identifikation der laufenden Instanz des Entities möglich.

3. KEY

   Beim *Key* handelt es sich um das Passwort für den jeweiligen Job. Hiermit ist ein Anmelden an den Server möglich.

4. MASTERID

   Die *MASTERID* ist die ID des Master Jobs, der den aktuellen Job ausführt.

5. PID

   Die *PID* ist die Prozess-ID auf Betriebssystemebene des Submitted Entities. Dies kann zum Beispiel für das Kill Program verwendet werden.

6. LOGFILE

   Der Parameter LOGFILE gibt das verwendete Logfile an, welches im Feld *Logfile* im Batches-and-Jobs-Dialog definiert wurde.

7. ERRORLOG

   Der Parameter ERRORLOG gibt das verwendete Error Logfile an, welches im Feld *Error Logfile* im Batches-and-Jobs-Dialog definiert wurde.

8. SDMSHOST

   Der Parameter SDMSHOST gibt den aktuellen Host des SDMS Servers an.

9. SDMSPORT

   Der Parameter SDMSPORT gibt den aktuellen Port des SDMS Servers auf dem Host an.

10. JOBTAG

    Das JOBTAG gibt das Child Tag des Submitted Entities an. Hiermit ist eine eindeutige Identifikation eines dynamisch erzeugten Child-Entities möglich.

11. SUBMITTIME

    Die SUBMITTIME gibt den Submitted-Zeitpunkt des Master Jobs an.

12. STARTTIME

    Die STARTTIME gibt den Zeitpunkt des tatsächlichen Starts des Submitted Entities auf dem Jobserver (falls es sich um einen Job handelt) an.

13. SEID

    Die SEID ist die ID des Scheduling Entities.

14. WORKDIR

    DER WORKDIR ist der Pfad des Working Directory.

15. JOBSTATE

    Der JOBSTATE ist der Exit State des Jobs, Batches oder Milestones.

16. MERGEDSTATE

    Der MERGEDSTATE ist der resultierende Exit State eines Jobs und seine Kinder.

17. EXPRUNTIME

    Die EXPRUNTIME ist die Expected Runtime wie in der Job Definition hinterlegt wurde.

18. EXPFINALTIME

    Die EXPFINALTIME ist die Expected Finaltime wie in der Job Definition hinterlegt wurde.

19. PARENTID

    Die PARENTID ist die ID des Parent Jobs Milestones oder Batches.

20. STATE

    Der STATE ist der aktuelle Status des Jobs.

21. ISRESTARTABLE

    ISRESTARTABLE sagt über den Job aus ob er Restartable ist oder nicht.

22. SYNCTIME

    Die SYNCTIME ist die Zeit zu der der Job in den Status SYNCHRONIZE_WAIT übergegangen ist.

23. RESOURCETIME

    Die RESOURCETIME ist die Zeit zu der der Job in den Status RESOURCE_WAIT übergegangen ist.

24. RUNNABLETIME

    RUNNABLETIME ist die Zeit zu der der Job in den Status RUNNABLE gewechselt ist.

25. FINISHTIME

    FINISHTIME ist der Endzeitpunkt des Prozesses.

26. SYSDATE

    Die Variable SYSDATE ist das aktuelle Datum und Uhrzeit.

27. LAST_WARNING

    Diese Variable enthält den Text des letzten Auditeintrags der das Hochsetzen des Warncounts zur Folge hatte.

28. RERUNSEQ

    Diese Variable gibt an wie oft der Job bereits restartet wurde.

29. SCOPENAME

    Liefert den vollen Pfadnamen des Jobservers, der den Job ausführt bzwàusgeführt hat.

Trigger-Parameter Für die Verwendung in Triggern gibt es eine weitere Anzahl von vordefinierten Parametern, welche in Skripten von Trigger-Jobs verwendet werden können. Diese Parameter stehen nur zur Verfügung, falls das Submitted Entity im Rahmen eines Triggers gestartet wurde. Wurde das Entity normal submitted sind diese Parameter nicht gefüllt.
Trigger-Umgebungs-Namen:

1. TRIGGERNAME

   Hierbei handelt es sich um den Namen des Triggers.

2. TRIGGERTYPE

   Hiermit kann aufgrund des Trigger-Typs ein Job unterschiedliche Handlungen vornehmen.

3. TRIGGERBASE

   Der Parameter TRIGGERBASE enthält den Namen des "Triggering Objects", also desjenigen Jobs, welcher zum Beispiel durch eine Exit State Translation den Trigger nach oben übermittelt.

4. TRIGGERORIGIN

   Der Parameter TRIGGERORIGIN gibt den Namen des Entities an, welcher den Trigger enthält. Das ist der Job der den Trigger in seinem Tab Trigger angezeigt bekommt.

5. TRIGGERREASON

   Der Parameter TRIGGERREASON gibt den Namen des Objektes aus, welcher den Trigger auslöst.

6. TRIGGERBASEID, TRIGGERORIGINID, TRIGGERREASONID

   Für jeden der drei oben genannten Objekte (TRIGGERBASE, TRIGGERREASON, TRIGGERORIGIN) wird in diesem Parameter die ID des Submitted Entities gespeichert.

7. TRIGGERBASEJOBID, TRIGGERORIGINJOBID, TRIGGERREASONJOBID

   Für jeden der drei oben genannten Objekte (TRIGGERBASE, TRIGGERREASON, TRIGGERORIGIN) wird in diesem Parameter die ID der Job Definition gespeichert.

8. TRIGGEROLDSTATE

   Der Parameter gibt den Job State vor dem auslösenden Moment, welches für den Trigger relevant ist, an.

9. TRIGGERNEWSTATE

   Der Parameter gibt den Job State nach dem auslösenden Moment, welches für den Trigger relevant ist, an.

10. TRIGGERSEQNO

    TRIGGERSEQNO ist die Anzahl der Auslösungen des Triggers.

14.5.11 Tab References

Im Tab References werden alle Referenzen (Parameter vom Type REFERENCE und CHILDREFERENCE) anderer Job bzw. Batchobjekte auf Parameter des aktuellen Objektes angezeigt.
Der Tab References sieht folgendermaßen aus:

14.5.12 Tab Trigger

Der Tab Trigger beschreibt alle Trigger die für dieses Scheduling Entity definiert sind. Ein Trigger dient zum Starten eines weiteren Ausführungsobjektes, falls beim aktuell gewählten Job (oder eines seiner Children, siehe unten) ein einstellbares Ereignis erreicht wurde. Als Beispiel können Benachrichtigungen verschickt werden,

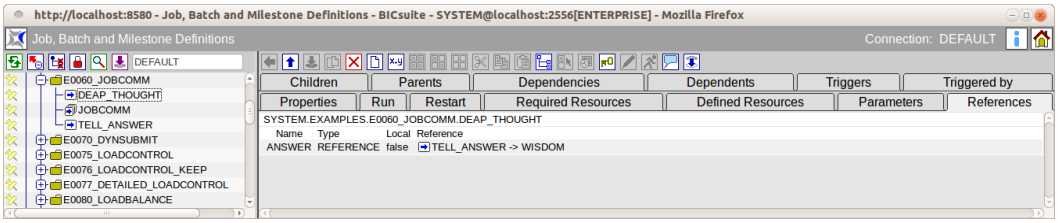


Abbildung 14.33: Batches und Jobs; References Tab

wenn ein Job beendet worden ist. Auch können Nachfolgeverarbeitungen oder eine Fehlerbehandlung bei Problemen automatisch gestartet werden.
Der Tab Trigger sieht folgendermaßen aus:

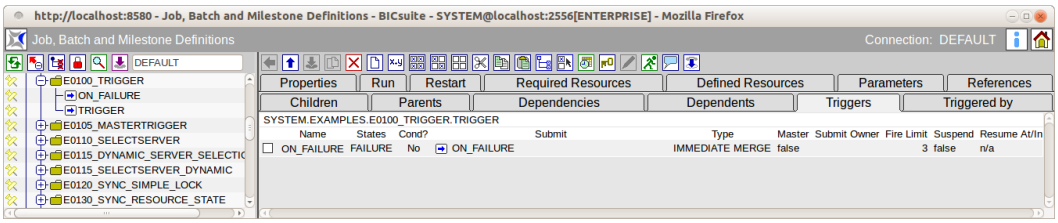


Abbildung 14.34: Batches und Jobs; Trigger Tab

Im Dialog "Trigger" wird eine Liste mit allen, für den gewählten Task, zugeordneten Triggern angezeigt. Durch Selektieren eines Jobs kann der Job, der durch den Trigger gestartet werden soll, mittels Copy und Paste in die Liste eingefügt werden. Der Name des Triggers wird beim Kopieren vom Namen des Jobs übernommen. Anschließend können durch anklicken des Trigger-Namens die Trigger Details angezeigt und geändert werden. Mittels des *Drop* Buttons werden Trigger gelöscht. Die Spalten der Liste haben folgende Bedeutung:

Name Das ist der Name des Triggers. Beim Einfügen eines neuen Jobs in die Trigger-Liste, wird der Trigger-Name auf dem Namen des Jobs gesetzt. Klickt man den Namen des Triggers an, kommt man in den Tab Trigger Details. Hier können alle Daten (einschließlich des Trigger-Namens) eingesehen und geändert werden.

States Hierbei handelt es sich um eine kommagetrennte Liste aller States, bei denen der Trigger beachtet werden soll.

Cond? Das Feld *Cond?* gibt an ob eine Bedingung spezifiziert wurde.

Submit Das ist der Name des Jobs oder Batches welcher durch den Trigger submitted werden soll.

Achtung: Dieses Feld ist ebenfalls durch einen Mausklick anwählbar. Wird dies gemacht, kommt man in die Editiermaske des jeweiligen Jobs. Hat man Daten hinzugefügt oder geändert und bestätigt die Sicherheitsabfrage, sind alle geänderten Daten verloren.
Die restlichen Spalten werden im nächsten Abschnitt erläutert.

14.5.12.1 Tab Trigger Details

Der Tab Trigger Details erscheint nachdem ein Trigger im Tab Trigger gewählt wurde. Im Trigger Details Tab können alle Detailinformationen bzgl. des Triggers geändert bzw. eingetragen werden.

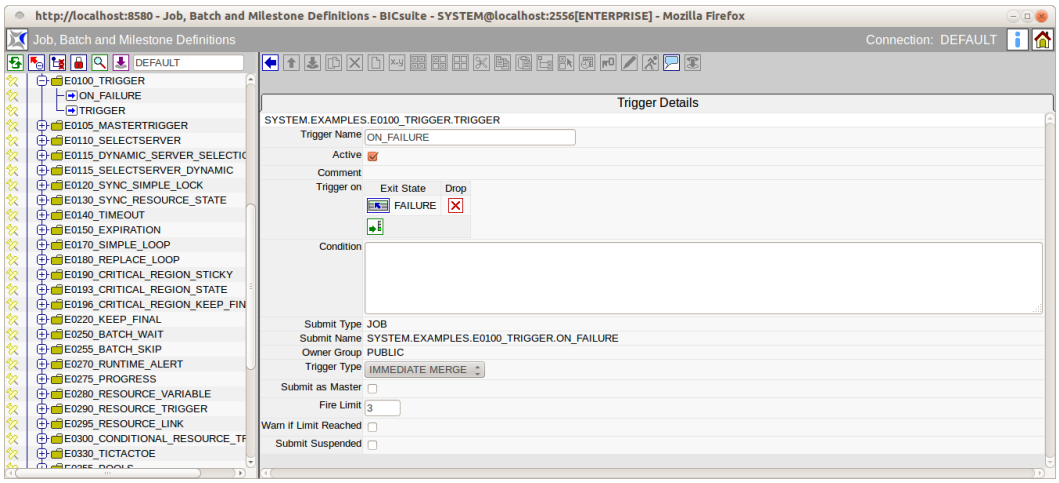


Abbildung 14.35: Batches und Jobs; Trigger Details

Folgende Felder sind auf dem Tab zu sehen:

Trigger Name Hierbei handelt es sich um den Namen des Triggers. Bei der Neuanlage eines Triggers, das heißt beim Einfügen eines Jobs in die Trigger-Liste, wird automatisch der Job Name übernommen. Anschließend kann in den Trigger Details der Name beliebig geändert werden.

Active Gibt an, ob der Trigger aktiv ist. Ist dieser Schalter nicht gesetzt, so wird der Trigger beim Eintreten des Trigger Ereignisses nicht ausgelöst. In der Liste der Trigger wird ein inaktiver Trigger mit einer schwarzen Hand (Suspend Symbol) angezeigt.

Liste Trigger On Hier kann eine Liste mit Exit States des gewählten Prozesses (nicht des zu triggernden Jobs) angegeben werden, bei denen der Trigger feuern

soll. Wird keine Liste angegeben, ist der Exit State des Jobs egal, er feuert immer wenn das zu erreichende Ereignis eingetreten ist.
Beispiel:
Ein Trigger soll starten wenn ein Job mit einem Fehler (EXIT_STATE: FAILURE) beendet wird. Hier muss die Liste Trigger On den State FAILURE gewählt haben. Andere States (SUCCESS, etc.) werden nicht eingetragen, da diese beim Feuern des Triggers nicht beachtet werden sollen.
Soll ein Trigger immer gestartet werden wenn der Prozess endet (unabhängig vom State, der vielleicht nur im Job, den der Trigger startet, unterschieden wird), so kann die Liste "Trigger on" leer gelassen oder mit allen möglichen States, die der Job annehmen kann, gefüllt werden.

Condition Im Feld *Condition* wird die Bedingung, die erfüllt werden muss, eingetragen.

Submit Type Das ist der Typ des Objektes (Batch oder Job) welches gestartet wird, wenn der Trigger feuert.

Submit Name Das ist der Pfad und der Name des Objektes welches gestartet wird, wenn der Trigger feuert. Hier handelt es sich um den kompletten Namen des Objektes mit allen übergeordneten Ordnerhierarchien

Owner Group Im Feld *Owner Group* ist der Eigentümer des Triggers definiert.

Trigger Type Hier handelt es sich um die Art des Ereignisses, welches den Trigger auslösen soll. Es gibt folgende Ereignisse:

1. Immediate Merge

   Der Job selber oder eines seiner Children hat einen State in der Trigger On-Liste angenommen. Ist kein Wert eingetragen, wird der Trigger bei jedem Statuswechsel gefeuert.

   Beispiel:

   Falls der Job oder eines seiner Children einen Fehler hat, soll eine SMS gesendet werden.

2. Immediate Local

   Der Job selber hat einen State in der Trigger On-Liste angenommen. Ist kein Wert eingetragen, wird der Trigger bei beenden des Jobs gefeuert. Der State der Children wird nicht berücksichtigt.

   Beispiel:

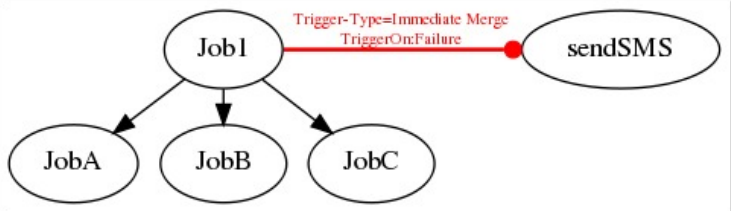


Abbildung 14.36: Beispiel für Immediate Merge Trigger

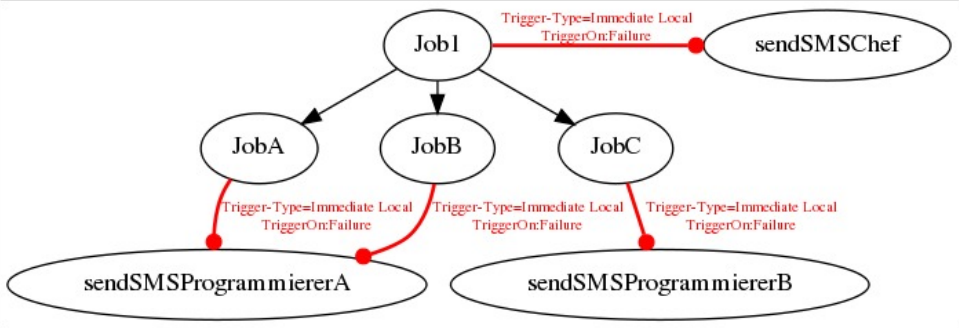


Abbildung 14.37: Beispiel für Immediate Local Trigger

Falls der Job einen Fehler hat, soll eine SMS an den Projektverantwortlichen gesendet werden. Für den Fall das einer der Children einen Fehler hat, wird im Child ein zweiter Trigger implementiert, der eine SMS an einen der verantwortlichen Programmierer schickt.

3. Before Final

   Die Trigger-Bedingung wird, kurz bevor das Scheduling Entity einen Final State erreichen würde, evaluiert. Die Trigger On-Liste muss hierbei nur aus Final State (oder keinen, dann wird auf jeden Final State getriggert) bestehen, da der Trigger sonst nicht feuert. Wird das zu triggernde Scheduling Entity nicht als Master submitted, kann der Scheduling Entity erst wieder Final werden, nachdem das durch den Trigger gestartete Scheduling Entity ebenfalls Final geworden ist.

   Im oberen Beispiel, wird JobC erneut aufgerufen, falls der Job fehlschlägt. Da dies ein Before Final Trigger ist, kann damit Job1 nicht Final werden, solange JobC ohne FAILURE beendet wurde.

4. After Final

   Der Trigger wird kurz nach dem der Job einen Final State erreicht hat ausgeführt. Die Trigger On-Liste sollte hierbei nur aus Final States bestehen (oder

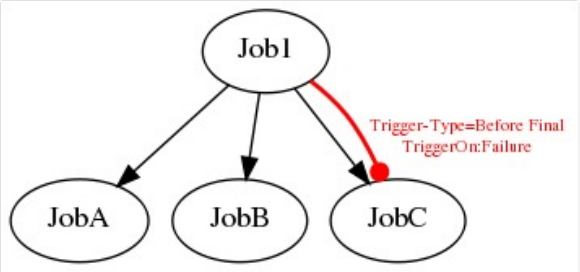


Abbildung 14.38: Beispiel für Before Final Trigger

keinen, dann wird auf jeden Final State getriggert), da der Trigger sonst nicht feuert.

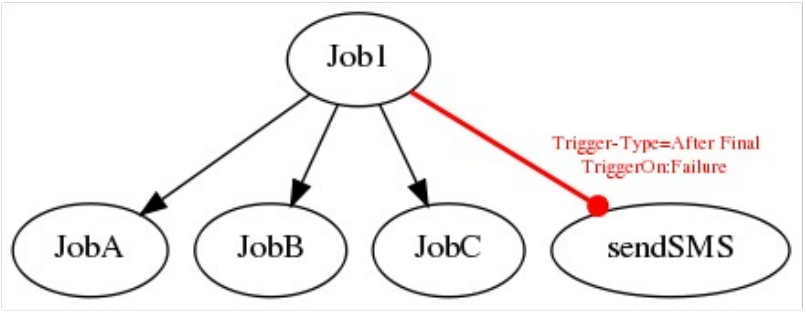


Abbildung 14.39: Beispiel für After Final Trigger

Die Legende für die Grafik finden Sie im Kapitel 1.8.

5. Finish Child

   Der Trigger wird ausgeführt, wenn ein direktes oder indirektes Child des triggernden Jobs einen State in der Trigger On-Liste erreicht. Hiermit ist es möglich eine Überwachung an zentraler Stelle zu implementieren.

   Im oberen Beispiel würde eine SMS (informiere Programmierer) geschickt werden, falls einer der Child-Jobs einen Failure State annimmt. Falls der eigene Job (Job1) erfolgreich ist, wird die SMS (großes Lob) geschickt.

6. Until Finished

   Der Until Finished Trigger überprüft periodisch die Condition. Wenn die Condition ”True” ist, wird solange getriggert, bis der State ”Finished” erreicht wird. Anschließend wird die Condition zum letzten Mal ausgewertet und wenn sie ”True” ist, wird getriggert.

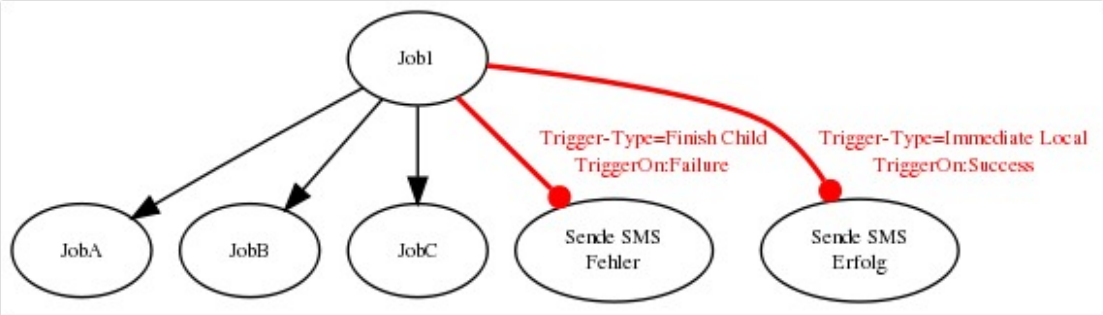


Abbildung 14.40: Beispiel für Finish Child Trigger

7. Until Final

   Der Until Final Trigger überprüft periodisch die Condition. Wenn die Condition ”True” ist, wird solange getriggert, bis der State ”Final” erreicht wird. Anschließend wird die Condition zum letzten Mal ausgewertet und wenn sie ”True” ist, wird getriggert.

8. Warning

   Diese Trigger werden gefeuert wenn der Warning-Flag für ein Job gesetzt wird.

Submit as Master Wurde das Flag Submit as Master gesetzt, so wird der Job, der durch den Trigger gestartet wird, als eigener Master Job submitted und hat keinen Einfluss auf den aktuellen Master Job-Lauf des triggernden Jobs. Wird das Flag nicht gesetzt, wird der getriggerte Job als Child des zu triggernden Jobs gestartet. Das bedeutet auch, dass der Job auf den getriggert wird, nicht beendet ist (einen Final State bekommen kann) solange der getriggerte Job noch läuft. Als einzige Ausnahme gilt der Trigger Type After Final bei dem der Trigger erst ausgelöst wird, nachdem der getriggerte Job Final geworden ist. Ist dies der Fall, wird der getriggerte Job als Child des Parents des triggernden Jobs gestartet. Wenn es sich bei dem getriggerten Job um das gleiche Scheduling Entity handelt wie der des triggernden Jobs, nimmt der getriggerte Job den Platz des triggernden Jobs ein.

Fire Limit Hiermit wird die Maximalanzahl der Auslösung des Triggers innerhalb eines Ablaufes angegeben. Ist die Maximalanzahl der Trigger-Auslösungen erreicht, so wird kein weiterer Trigger mehr ausgelöst und kein neuer Job mehr gestartet.

Warn if Limit Reached Bei der Anlage eines Triggers kann festgelegt werden, dass ein Job eine Warnung bekommt (Warn-Flag gesetzt, Eintrag in Audit) wenn das Limit erreicht wird. Dies gilt nur für Job-Trigger. Per Default wird keine Warnung ausgegeben.

Submit Suspended Gibt an ob das getriggerte Scheduling Entity als Suspended gestartet werden soll.

Resume Wird nur angezeigt falls *Submit Suspended* gesetzt wurde.
Hier kann ausgewählt werden, ob ein automatischer Resume stattfinden soll.
Es gibt folgende Möglichkeiten:

- NO: wählt diese Funktionalität ab.
- AT: wählt einen automatischen Resume zu einem festen Zeitpunkt. Das Eingabefeld *Resume Time* wird angezeigt.
- IN: wählt einen automatischen Resume nach Ablauf einer Zeit. Die Eingabefelder *Resume In* und *Unit* werden angezeigt.

Resume Time Wird nur angezeigt falls im *Resume* "AT" gewählt wurde.
Hier wird der gewünschte Resume Zeitpunkt im Format "YYYY-MM-DDTHH:MI:SS" eingegeben.
Das Format orientiert sich an der ISO Norm 8601 und erlaubt auch unvollständige Angaben. Die Eingabe von 'T16:00' wird den Job um 16:00 Uhr resumen (ausgehend von der Zeit der Triggerauslösung).

Resume In Wird nur angezeigt falls im *Resume* "IN" gewählt wurde.
Hier wird angegeben, wie viele Zeiteinheiten (siehe *Unit*) bis zum Resume gewartet werden soll.

Unit Wird nur angezeigt falls im *Resume* "IN" gewählt wurde.
Hier wird eingegeben ob es sich bei der Eingabe im *Resume In* um Minuten (MINUTE), Stunden (HOUR), oder Tage (DAY) handeln soll.

14.5.13 Tab Triggered by

Der Tab "Triggered by" listed alle Trigger, welche dieses Scheduling Entity submitten.
Der Tab "Triggered by" sieht folgendermaßen aus:
Die Spalten der Liste haben folgende Bedeutung:

Name Das ist der Name des Triggers.

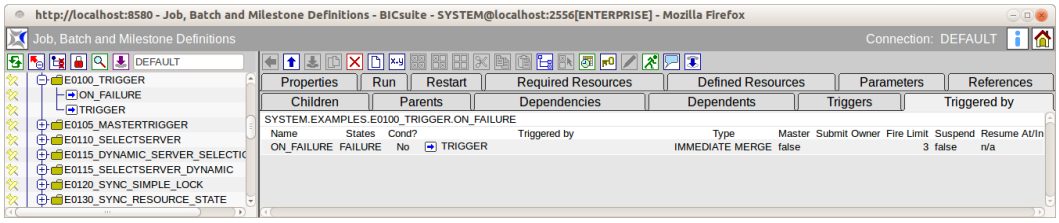


Abbildung 14.41: Jobs und Batches; Triggered By Tab

States Hierbei handelt es sich um eine kommagetrennte Liste aller States, bei denen der Trigger beachtet werden soll.

Cond? Das Feld *Cond?* gibt an ob eine Bedingung spezifiziert wurde.

Triggered by Das ist der Name des Jobs, Batches oder Milestones, welcher den Trigger definiert.
Achtung: Dieses Feld ist ebenfalls durch einen Mausklick anwählbar. Wird dies gemacht, kommt man in die Editiermaske des jeweiligen Jobs. Hat man Daten hinzugefügt oder geändert und bestätigt die Sicherheitsabfrage, sind alle geänderten Daten verloren.
Die Bedeutung der restlichen Spalten sind dem Kapitel "Tab Trigger Details" zu entnehmen.

14.6 Job Hierarchy Navigation

Das Navigationsfenster Job Hierarchy kann im Dialog "Batches and Jobs" zu jedem Batch und Job angezeigt werden. Der Navigationsbildschirm sieht folgendermaßen aus:

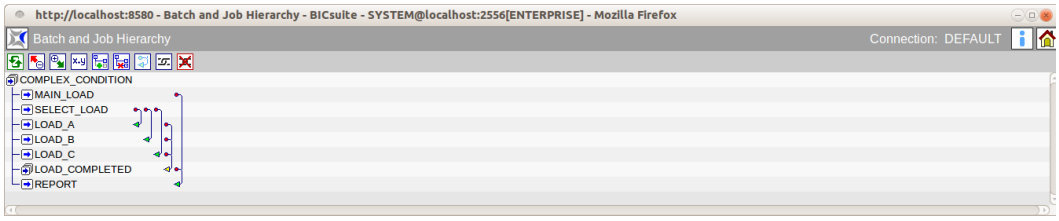


Abbildung 14.42: Batches und Jobs; Hierarchie Ansicht

Im Navigationsfenster werden alle Children des aktuellen Entities angezeigt. Haben die Children auch eine Child-Hierarchie, so kann diese ebenfalls expandiert werden, bis die komplette Hierarchie des Objektes sichtbar ist.
Zusätzlich zu den Standard-Buttons sind folgende Buttons sichtbar:

 Add Children

Mit dem *Add Children* Button, kann man Children direkt in das Job Hierarchy Navigationsfenster einfügen. Dazu wird der Button gedrückt und anschließend der Job oder Batch, der die Children bekommen soll, angeklickt. Es erscheint die Maske, in der ein oder mehrere Children ausgewählt werden können. Nachdem Sie ausgewählt haben, können Sie die Maske wieder schließen und als letztes wird der Parent nochmals angeklickt und die Children werden automatisch hinzugefügt.

 Remove Child

Mit dem *Remove Child* Button, können die Children gelöscht werden. Dazu wird der Button gedrückt und anschließend der Parent. Nachdem Sie dies mit "OK" bestätigt haben, werden die Children automatisch gelöscht.

 Show Dependencies

Der *Show Dependencies* Button, ist ein Schalter mit zwei Werten. Standardmäßig werden die Abhängigkeiten im Dialog nicht angezeigt und es erscheint der *Show Dependencies* Button in der Buttonleiste.
Mit dem *Show Dependencies* Button werden die Abhängigkeiten der Scheduling Entities untereinander graphisch angezeigt. Jede Abhängigkeit wird mittels eines Pfeils angezeigt. Das benötigte Scheduling Entity (required Scheduling Entity) stellt den Start des Pfeils dar (der runde Teil des Pfeils) und das abhängige Scheduling Entity (dependent Scheduling Entity) ist die Pfeilspitze.
Beispiel:

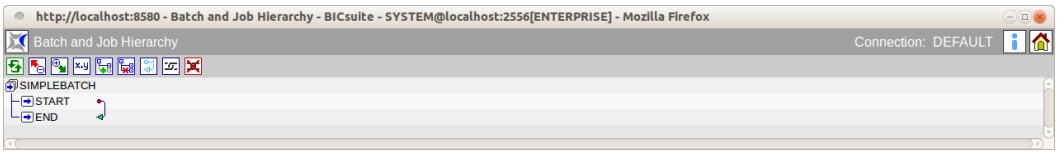


Abbildung 14.43: Hierarchie Ansicht mit Dependencies

Im oberen Beispiel ist der Job END vom Job START abhängig.
Ist ein Job von mehreren Objekten abhängig, so erscheint ein Pfeil mit mehreren Anfängen (die runden Teile des Pfeils), welche alle benötigten Scheduling Entities (required Scheduling Entities) darstellen. Die Pfeilspitze bleibt weiterhin das abhängige Scheduling Entity.
Beispiel:
In diesem Beispiel ist der Batch LOAD_COMPLETED von den Jobs LOAD_1, LOAD_2 und LOAD_3 abhängig.

 Hide Dependencies

Der *Hide Dependencies* Button ist der zweite Zustand des Schalters. Er erscheint falls der *Show Dependencies* Button gedrückt wurde und dadurch die Anzeige der

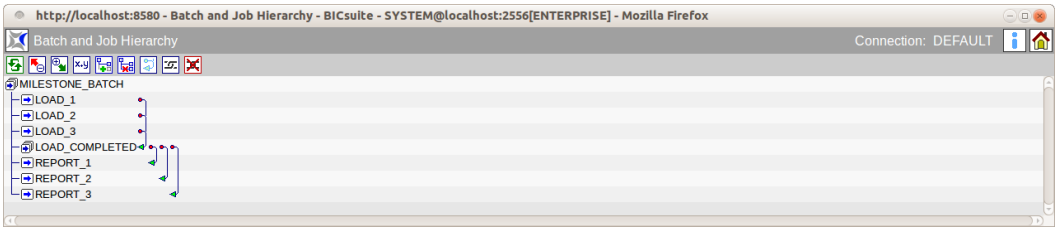


Abbildung 14.44: Hierarchie Ansicht mit mehrfachen Dependencies

Abhängigkeiten aktiviert ist. Mittels des *Hide Dependencies* Buttons ist es nun möglich die Anzeige der Abhängigkeiten wieder abzuschalten.

 Chain

Mit dem *Chain* Button kommt man in den Chain Modus. Hier kann man die Abhängigkeiten von Jobs direkt im Job Hierarchy Navigationsbildschirm einfügen. Dazu wird als erstes der *Chain* Button angeklickt und anschließend der Dependent Job und Required Job. Jetzt besteht eine Abhängigkeitsbeziehung zwischen den beiden Jobs.
Mit dem *Cancel* Button verlässt man den Chain Modus.

 Unchain

Mit dem *Unchain* Button werden Abhängigkeitsbeziehungen gelöst. Dazu wird als erstes der *Unchain* angeklickt und anschließend die Abhängigkeit die gelöst werden soll.
Mit den *Chain/Unchain* Buttons ist es einfach zyklische Abhängigkeiten zu erzeugen. Sie werden vom Frontend festgestellt und gemeldet. Siehe dazu auch Bild

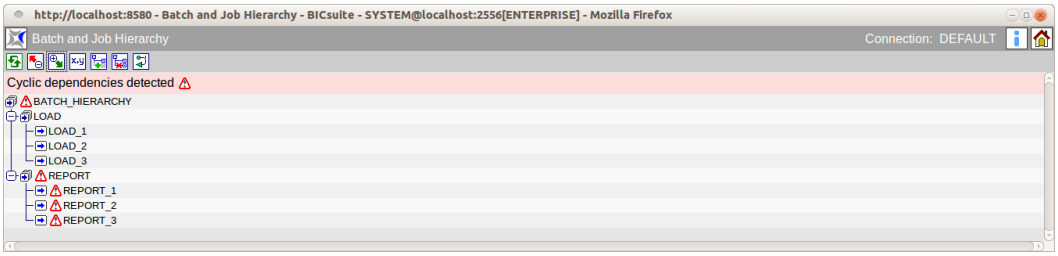


Abbildung 14.45: Zyklische Abhängigkeiten

14.45.

15 BICsuite!Web Users

15.1 Bild

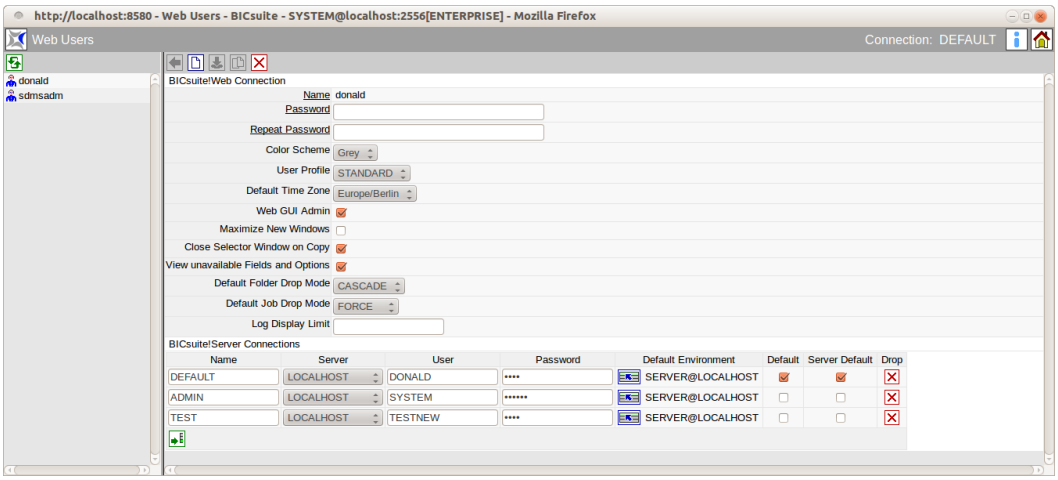


Abbildung 15.1: Benutzerverwaltung für das Web Frontend

15.2 Konzept

15.2.1 Kurzbeschreibung

Die Benutzerverwaltung dient der Definition von BICsuite Oberflächenbenutzern. Jeder Benutzer, welcher mit der BICsuite Oberfläche arbeiten will, sollte eine eigene Benutzerkennung haben. Der Benutzer muss sich während der Anmeldung mit dieser Benutzerkennung und deren Passwort anmelden.

15.2.2 Ausführliche Beschreibung

Zusätzlich zu Darstellungsoptionen und Sicherheitsmechanismen, ist in diesem Dialog die Verbindung zum Server pflegbar. Der BICsuite Server verwendet eine eigene Benutzerverwaltung, welche unabhängig von der Benutzerverwaltung des Oberflächensystems ist. Aus diesem Grund ist es möglich, dass alle Oberflächenbenutzer

nur einen Serverbenutzer benutzen oder dass eine 1:1 Abbildung der Oberflächenbenutzer auch im Server erfolgt. Bitte fragen Sie bei Änderungen an dem Benutzeraccounts immer bei Ihrem Systemadministrator nach.

15.3 Navigator

Im Navigationsbildschirm werden alle Benutzerkennungen angezeigt.

15.4 Editor

Im Editor werden alle Daten eines gewählten Benutzers gepflegt.

15.4.1 BICsuite!Web Connection

Die Felder im Web-GUI-Connect haben folgende Bedeutung:

Name Der Anmeldename des Benutzers.
Der Anmeldename ist im Dialog Login als Benutzername einzugeben.

Password Das *Password* dient zur Authentifizierung jedes Benutzers, welcher sich am BICsuite System anmeldet und muss beim Login als Passwort eingegeben werden.
Das Passwort wird in der Anzeige verdeckt und muss identisch mit dem Repeat Password sein.
Da das Passwort im Dialog nicht angezeigt wird, ist es nicht ersichtlich ob ein Passwort bereits eingegeben wurde oder dies noch notwendig ist. Er muss allerdings nur einmal eingegeben werden.

Repeat Password Das *Repeat Password* muss bei der Eingabe des Passwortes identisch zum Password sein.
Die doppelte Eingabe ist nötig, da ein Tippfehler beim blinden Eingeben des Passwortes zu Problemen führen kann. Da es unwahrscheinlich ist, dass zweimal der selbe Tippfehler gemacht wurde, geht man davon aus, dass bei einer korrekten zweifachen Eingabe des Passwortes, der gewollte Begriff richtig eingetragen wurde.

Color Scheme Hier wird das für das Interface verwendete Farbschema ausgewählt.

User Profile Hier wird das Profil des Users eingegeben.

Default Time Zone Dieser Parameter definiert welche Zeitzone bei der Neuanlage von Time Schedules voreingestellt wird. Weiterhin werden alle Zeitstempel in dieser Zeitzone angezeigt.

Time Stamp Format Hier kann ein vom Default (%Y.%m.%d %H:%M:%S Beispiel 2014.06.18 14:06) abweichendes Format für die Anzeige von Zeitstempeln eingestellt werden. Das Format entspricht dem der strftime C Standardfunktion.

Web-GUI-Admin Handelt es sich bei diesem Benutzer um einen Administrator der Web-Oberfläche, muss dieses Häkchen gesetzt werden.

Maximize New Windows Neue Dialoge werden immer maximiert angezeigt, wenn das Häkchen gesetzt wurde. Ist es nicht gesetzt, erscheinen die Fenster immer in der normalen Größe.

Close Selector Window on Copy Ist das Häkchen gesetzt, werden Fenster welche zu Selektion von Objekten geöffnet werden nach dem Betätigen des "Copy" Buttons automatisch geschlossen.

View unavailable Fields and Options Ist das Häkchen gesetzt, werden die nicht zur Verfügung stehenden Felder und Optionen angezeigt.

Default Folder Drop Mode Hierbei handelt es sich um den Modus der verwendet wird, wenn ein Ordner im Dialog Batches and Jobs gelöscht werden soll. Es gibt folgende Optionen:

1. NORMAL

   In diesem Modus ist es nur möglich einen Ordner zu löschen wenn er leer ist, also keine Unterordner und keine Jobs, Batches oder Milestones besitzt.

   Dies ist der "Default" Wert des Systems.

2. CASCADE

   In diesem Modus wird der Ordner kaskadierend gelöscht. Es werden alle Unterordner und alle darunterliegenden Jobs, Batches und Milestones gelöscht.

   Achtung: Diese Einstellung ist mit großer Vorsicht zu genießen, da es damit möglich ist, viele Objekte auf einmal zu löschen, obwohl man dies gar nicht beabsichtigt hat.

Default Job Drop Mode Hierbei handelt es sich um den Modus welcher verwendet wird, wenn ein Job bzw. Batch oder Milestone gelöscht werden soll.
OPTIONEN

1. NORMAL

In diesem Modus ist es nur möglich einen Job, Batch oder Milestonzu löschen, falls keine anderen Jobs mehr Abhängigkeiten zu diesem Job haben. Auch darf das Scheduling Entity keine Child-Beziehungen mehr besitzen.

Dies ist der "Default" Wert des Systems.

2. FORCE

   In diesem Modus wird das Scheduling Entity gelöscht, obwohl noch andere Scheduling Entities Abhängigkeiten auf ihn besitzen und obwohl er noch Child-Beziehungen hat.

15.4.2 BICsuite!Server Connections

Hier kann eine oder mehrere Server Verbindung konfiguriert werden.
Auf dem Main Desktop kann über ein Options Feld in der Kopfzeile gewählt werden, welche Server Verbindung für das nächste zu öffnende Fenster verwendet werden soll.
Nach dem Editieren der Server Verbindungen muss evtl. der Main Desktop im Browser neu geladen werden, damit die Server Verbindungen im Optionsfeld der Kopfzeile richtig angezeigt werden.

Name Der Name der Server Verbindung, für die Auswahl in der Kopfzeile des Main Desktops.

Server Der Name des Servers wie er vom BICsuite!Web Systemadministrator in /Custom/SDMSServers (in Zope) eingetragen wurde.

User Der Name des BICsuite Server Benutzeraccounts. Dieser kann unterschiedlich zum Namen der Web Users sein.

Password Mit dieser Benutzerkennung und dem Password verbindet sich die BICsuite Oberfläche mit dem BICsuite Server.
Das Passwort wird in der Anzeige verdeckt.

Default Environment Das Default Environment wird bei Jobs standardmäßig eingetragen.

Default Ist das Häkchen gesetzt, so wird diese Server Verbindung als Grundeinstellung verwendet. Es muss für genau eine Zeile gesetzt werden.

Server Default Ist das Häkchen gesetzt, so wird diese Server Verbindung als Grundeinstellung verwendet falls nur der Server bekannt ist. Es muss für jeden Server in genau einer Zeile gesetzt werden.

16 BICsuite Server Users

16.1 Bild

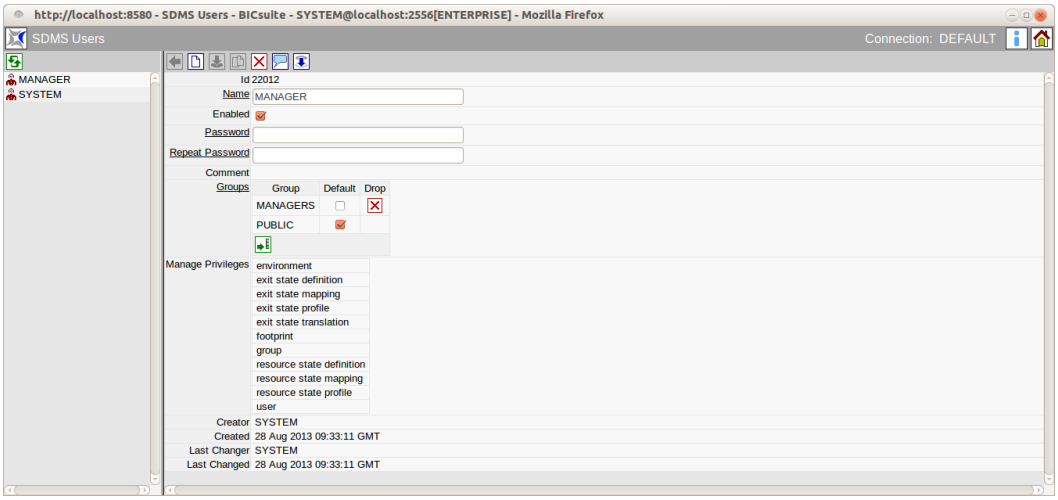


Abbildung 16.1: BICsuite Server Benutzerverwaltung

16.2 Konzept

16.2.1 Kurzbeschreibung

Der BICsuite Server User ist für die Legitimierung beim Scheduling Server notwendig.

16.2.2 Ausführliche Beschreibung

Da die Bedienung des Scheduling Servers nicht ausschließlich über die in dieser Dokumentation beschriebene Web-Oberfläche, sondern auch mit Hilfe anderer Schnittstellen wie etwa *sdmsh* erfolgen kann, ist eine Zugangskontrolle im Server notwendig. Die Pflege der Benutzer bzw. ihre Gruppenzugehörigkeit kann mittels des hier beschriebenen Dialogs durchgeführt werden.

16.3 Navigator

Im Navigationsbildschirm werden alle Benutzer angezeigt.
Der Navigatorbereich wird nur angezeigt wenn der aktive Benutzer Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage user” Privileg ist. Allen anderen Benutzern wird nur der Editor angezeigt über welchen diese ihr eigenes Passwort und die als Grundeinstellung zu wählende Gruppe ändern können.

16.4 Editor

Im Editor werden alle Daten des gewählten oder neu angelegten Benutzers gepflegt. SDMS User dürfen nur von Benutzern, welche Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage user” Privileg sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder ”read only”.
Folgende Felder sind im Editor zu sehen:

Name Der Anmeldename des Benutzers.
Der Anmeldename ist im Dialog Login als Benutzername einzugeben.

Enabled Ist der aktive Benutzer Mitglied der Gruppe ”ADMIN” oder einer anderen Gruppe mit dem ”manage user” privileg, so kann hier festgelegt werden ob ein Login dieses Benutzers erlaubt werden soll.

Password Das *Password* dient zur Authentifizierung jedes Benutzers, welcher sich am BICsuite System anmeldet und muss beim Login als Passwort eingegeben werden.
Das Passwort wird in der Anzeige verdeckt und muss identisch mit dem *Repeat Password* sein.
Da das Passwort im Dialog nicht angezeigt wird, ist es nicht ersichtlich ob ein Passwort bereits eingegeben wurde oder dies noch notwendig ist. Er muss allerdings nur einmal eingegeben werden.
Die maximale Länge eines Passwortes beträgt 64 Zeichen. Alle Zeichen, also auch Leerzeichen, sind zulässig.
Das Passwort kann immer geändert werden wenn der aktive Benutzer Mitglied der Gruppe ”ADMIN” ist oder der zu ändernde Benutzer der aktive Benutzer ist (man kann natürlich sein eigenes Passwort ändern).
Ist der aktive Benutzer Mitglied einer anderen Gruppe mit dem ”manage user” Privileg, so darf er das Passwort nur dann ändern, wenn er selbst Mitglied aller Gruppen des zu ändernden Benutzers ist.
Damit wird sichergestellt, dass ein Benutzer mit ”manage user” Privileg seine eigenen Zugriffsrechte nicht erweitern kann.

Repeat Password Das *Repeat Password* muss bei der Eingabe des Passwortes identisch zum Password sein.
Die doppelte Eingabe ist nötig, da ein Tippfehler beim blinden Eingeben des Passwortes zu Problemen führen kann. Da es unwahrscheinlich ist, dass zweimal der selbe Tippfehler gemacht wurde, geht man davon aus, dass bei einer korrekten zweifachen Eingabe des Passwortes, der gewollte Begriff richtig eingetragen wurde.

Groups In der Tabelle "Groups" sind die Gruppen zu denen der Benutzer angehört aufgeführt.
Es gibt folgende Spalten:

Group Gruppen zu denen der angegebene User-Mitglied gehört.

Default Der Default zeigt an zu welcher Gruppe ein neu angelegtes Objekt gehört, wenn keine Gruppe spezifiziert ist.

Drop Dieser Button dient zum Löschen der Gruppenzugehörigkeit. Vor dem Löschen erfolgt eine Sicherheitsabfrage.

Manage Privileges Hier werden die Manage Privilegien welche der User über seine Gruppenzugehörigkeit besitzt angezeigt.

17 Groups

17.1 Bild

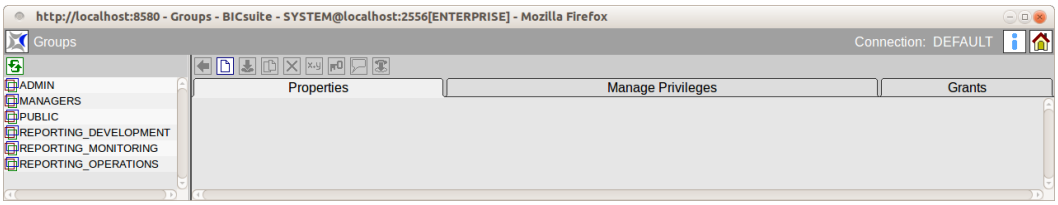


Abbildung 17.1: Gruppenverwaltung

17.2 Konzept

17.2.1 Beschreibung

Die Gruppen sind die Rechtenträger. Die einzelnen Benutzer gehören Gruppen an und sie haben die Rechte die den Gruppen gehören. Ein User kann mehreren Gruppen angehören. Folgende Gruppen sind immer vorhanden:

- Public: Jeder Benutzer ist immer ein Mitglied der Gruppe *Public*.
- Admin: Jeder Benutzer der Gruppe *Admin* hat alle Rechte im System.

17.3 Navigator

Im Navigator stehen alle sichtbare Gruppen in einer Liste. Sichtbar in diesem Kontext bedeutet, dass man entweder Mitglied der Gruppe ist, oder aber das View Privileg für die Gruppe hat.

17.4 Editor

17.4.1 Tab Properties

Dieser Tab dient der Pflege der Gruppeneigenschaften.
Folgende Felder sind im zu sehen:

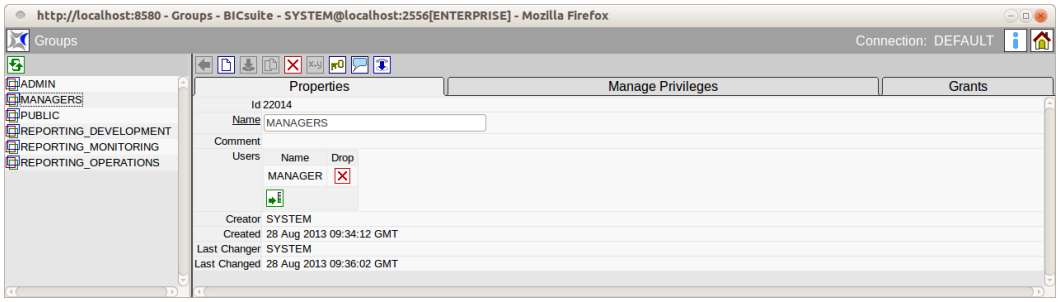


Abbildung 17.2: Gruppeneigenschaften

ID In diesem Feld wir die Group ID angezeigt.

Name Im Feld *Name* steht der Name der Gruppe.

Users Folgendes Feld steht in der Liste Users.

Name Im Feld *Name* steht der Benutzer der Gruppe.

17.4.2 Tab Manage Privileges

Der Manage Privileges Tab ermöglicht die Verwaltung der Manage Privilegien der Gruppe. Manage Privileges dürfen nur von Benutzern, welche Mitglied der Gruppe ”ADMIN” sind, editiert werden. Für alle anderen Benutzer sind alle Eingabefelder ”read only”.

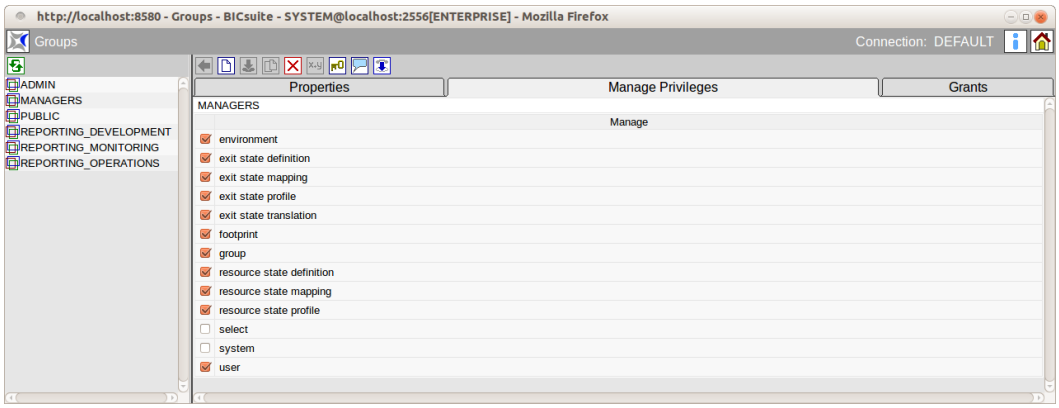


Abbildung 17.3: Manage Privilegien einer Gruppe

Er sieht aus wie im Bild 17.3.

17.4.3 Grants

Der Grants Tab zeigt alle Objekte an für die der Gruppe Privilegien zugeordnet wurden.
Der Objekttyp ist am jeweiligen Icon erkennbar.
Die Liste wird immer nach Objektname sortiert.
Wird der Mauszeiger über das Privilegienfeld einer Zeile gebracht, so erscheint eine Popup Info, welche den Privilegien String erklärt.

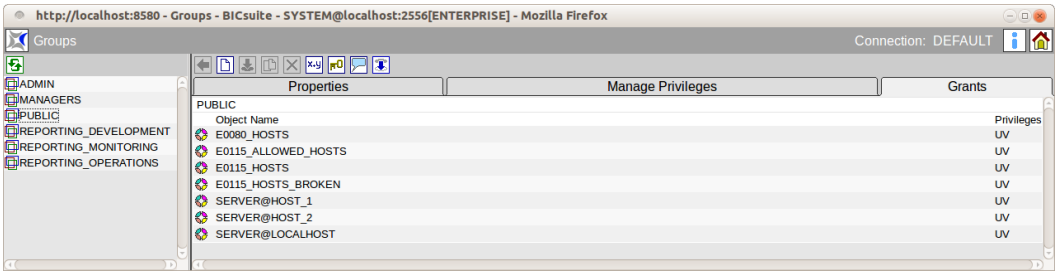


Abbildung 17.4: Objektprivilegien einer Gruppe

Er sieht aus wie im Bild 17.4.

Editor

18 Time Scheduling

18.1 Bild

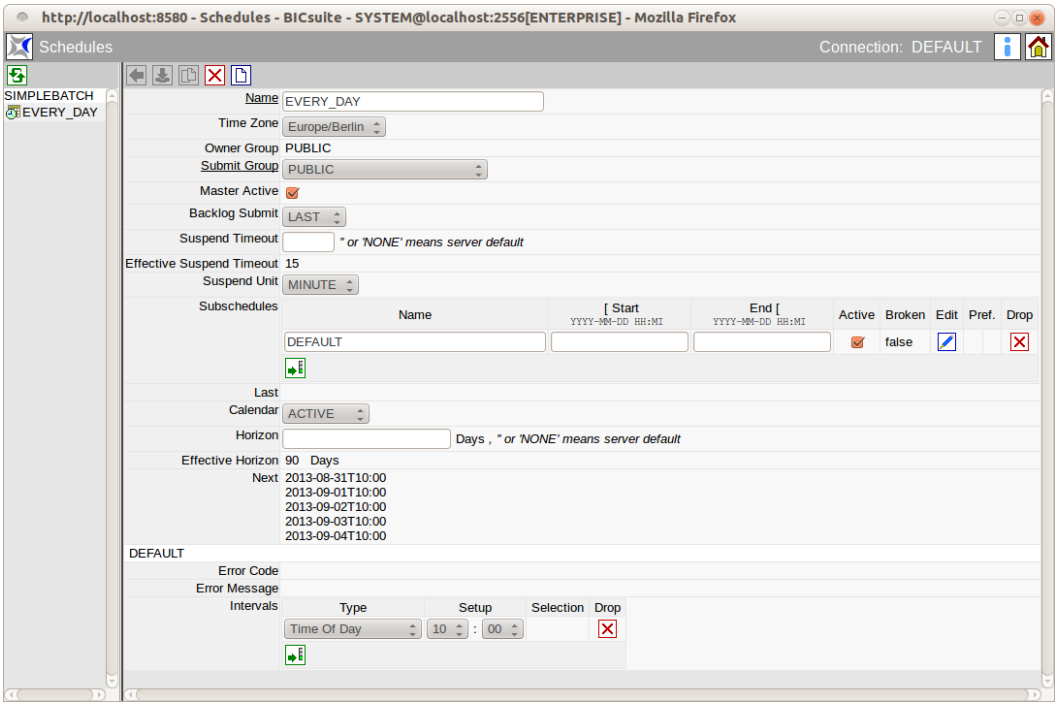


Abbildung 18.1: Time Scheduling

18.2 Konzept

18.2.1 Kurzbeschreibung

Das Time Scheduling erlaubt das Starten von Batches und Jobs zu bestimmten Zeitpunkten. Das Time Scheduling erlaubt eine einfache Definition komplexer Zeitpläne für Jobs und Batches.
Eine Anzahl von Zeitpunkten, die angeben wann der Job gestartet werden soll, nennt man einen Zeitplan (Schedule).
Zeitpläne gelten immer nur für einen Master Job.

18.2.2 Ausführliche Beschreibung

Im Dialog "Time Scheduling" wird zwischen Main Schedules, welche in der Navigation als Elemente angezeigt werden und Sub Schedules, welche innerhalb eines Main Schedules verwaltet werden können, unterschieden.
Main Schedules sind untereinander unabhängig und es findet keine Interaktion statt.
Sub Schedules können vom Zeitintervall getrennt oder überschneidend definiert werden. Damit können verschiedene Ausfallzeiten simuliert oder unterschiedliche Zeitpläne eingestellt werden, die damit immer aktiv sind.
Falls auf Zuruf die Schedules geändert werden sollen, kann dies als Interimszeitplan eingegeben werden. Um den Interimsplan zu aktivieren, muss ihm die höchste Priorität zugeteilt werden. Der ursprüngliche Zeitplan bleibt dabei erhalten. Der ursprüngliche Zeitplan wird wieder aktiviert, indem man ihm wieder die höchste Priorität zuteilt.

18.3 Navigator

Die Navigation zeigt die für dieses Scheduling Entity vorhandenen Zeitpläne (Main Schedules) an. Ein Scheduling Entity kann mehrere Main Schedules besitzen. Diese Main Schedules werden unabhängig berücksichtigt. Das heißt, es werden die Startzeiten aller Main Schedules berücksichtigt.
Wenn mehrere Main Schedules identische Startzeiten vorsehen, wird das Scheduling Entity mehrfach gestartet.

18.4 Editor

Der Editor zeigt, für den in der Navigation gewählten Main Schedule, alle "Detail" Elemente an. Hier können Sub Schedules angelegt und eine Liste von Zeitpunkten gepflegt werden, an denen der gewählte Job oder Batch gestartet werden soll. Die "Editor" Maske sieht folgendermaßen aus:
Die obigen Felder haben folgende Bedeutung:

Name Das ist der Name des Main Schedules. Beim ersten Anlegen eines neuen Main Schedules wird dieser automatisch mit "MASTER" vorbelegt. Er ist aber beliebig änderbar. Sollten mehrere Main Schedules angelegt werden, ist jeweils ein unterschiedlicher Name anzugeben.

Time Zone Die Zeitzone für die der Schedule berechnet werden soll. Welche Zeitzonen hier zur Verfügung stehen, kann vom BICsuite!web Administrator konfiguriert werden.

Owner Group Im Feld *Owner Group* ist der Eigentümer des Schedules definiert.

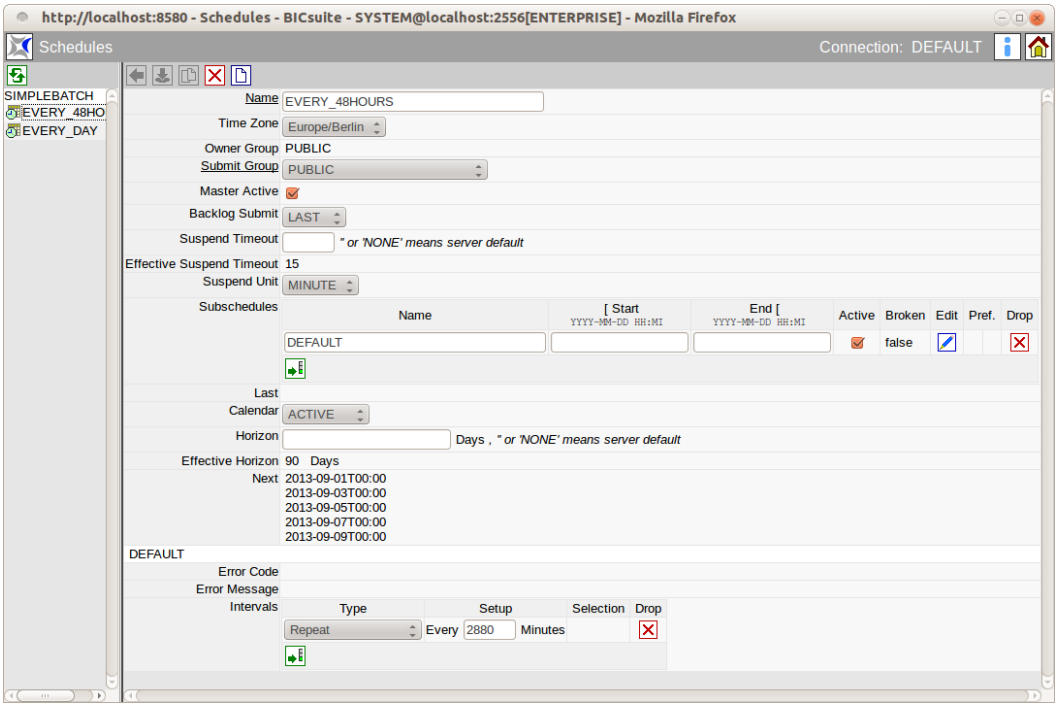


Abbildung 18.2: Time Scheduling Navigator

Submit Group Die *Submit Group* bestimmt die Eigentümergruppe des Jobs der submitted wird.

Master Active Ist das Flag im Master Active gesetzt, wird der Schedule vom Time Scheduling System berücksichtigt, wenn nicht, wird er ignoriert.

Backlog Submit Der Backlog bestimmt wie sich das Time Scheduling System nach einem Serverdowntime verhält:

- None: Die ”versäumten” Jobs werden nicht mehr ausgeführt.
- Last: Es wird nur der letzte der ”versäumten” Jobs ausgeführt.
- All: Es werden alle ”versäumten” Jobs ausgeführt.

Suspend Timeout, Suspend Unit Ein Job wird nach einem Serverdowntime suspended submitted, wenn die geplante Submit-Zeit länger als die eingegebene Zeit in den Feldern *Suspend Timeout* und *Suspend Unit* her ist. Wenn keine Zeit eingetragen ist, gilt der serverweite Default.

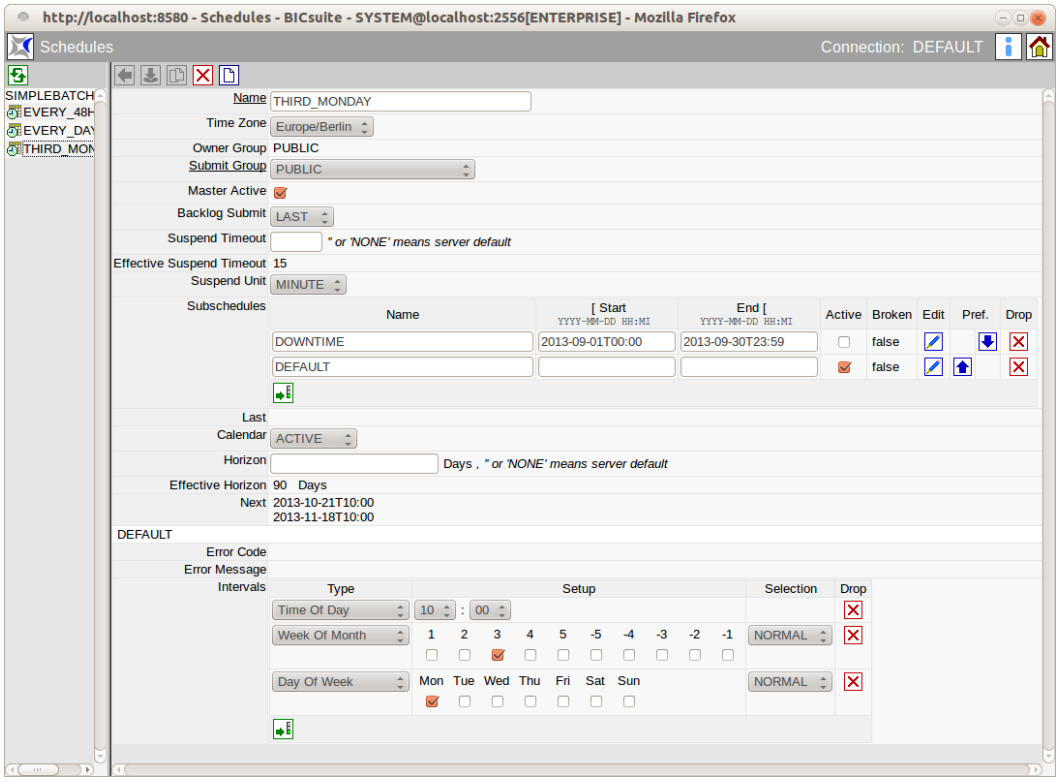


Abbildung 18.3: Time Scheduling Editor

Liste Sub Schedules In der Liste ”Sub Schedules” werden alle Sub Schedules des Main Schedules angezeigt. Die Schedules sind nur in dem eingegebenen Zeitraum gültig. Die Sub Schedules mit der höchsten Präferenz (zu oberst stehende) werden als erstes berücksichtigt. Diese ’Überdecken’ für Ihren Zeitraum weiter unten stehende Sub Schedules. Die bedeutet, dass zu einem Zeitpunkt immer nur genau ein Sub Schedule gültig sein kann.
Die Felder der Liste haben folgende Bedeutung:

Name Im Feld *Name* wird der Name des Sub Schedules eingetragen.

Start Im Feld *Start* wird das Datum und die Uhrzeit eingetragen, ab dem dieses Sub Schedule gültig ist. Wird das Feld leer gelassen, so gilt der Subschedule als ”schon immer gültig”.

End Im Feld *End* wird das Datum und die Uhrzeit eingetragen, ab dem dieses Sub Schedule ungültig wird. Wird das Feld leer gelassen, so gilt der Subschedule als ”ewig gültig”.

In der Liste der Sub Schedules stehen folgende Buttons zur Verfügung:

Active Nur wenn das Flag gesetzt ist, werden die Jobs submitted. Durch löschen des Flags kann das Submitten folglich verhindert werden. Downtimes werden typischer Weise als hoch präferenzielle Sub Schedules mit nicht gesetztem Active Flag implementiert.

Broken Mittels des Broken Feldes kann geprüft werden, ob beim Submit des Jobs ein Fehler aufgetreten ist. Ist dies der Fall, wird bei diesem Sub Schedule das Feld *Broken* mit TRUE angegeben.

Edit Nach Betätigung des *Edit* Buttons, erscheint die zu dem Sub Schedule gehörende Intervallbeschreibung, in der man präzise festlegen kann wann der Job gestartet wird.

Pref. Über die Preference Buttons lässt sich die Präferenz der einzelnen Sub Schedules festlegen. Dazu werden die Zeilen nach betätigen der Buttons nach oben oder unten verschoben.

Drop Dieser Button dient zum Löschen des Objektes.

Last Im Feld *Last* wird der letzte Ausführungszeitpunkt des Jobs durch das Scheduling System angezeigt. Wurde der Job durch das Scheduling System noch nie gestartet, ist dieses Feld leer.
Das Datumsformat sieht folgendermaßen aus:
YYYY:MM:DDTHH:MM
Die Platzhalter entsprechen denen im Feld *Start*.

Calendar Dieses Feld bestimmt, ob die geplante Ausführungszeiten des Jobs in den Kalender aufgenommen werden. Ist dies der Fall, haben auch die beide nächste Felder eine Bedeutung.

Horizon Das Feld Horizon bestimmt wie weit im Voraus Kalendereinträge gemacht werden sollen. Die Zeitangabe erfolgt in Tagen.

Effective Horizon Falls keinen Eintrag in das Feld Horizon gemacht wird, gilt den konfigurierten Defaultwert als Horizont. Das Feld Effective Horizon zeigt den aktuell gültigen Wert.

Next Im Feld *Next* wird der oder die nächste geplante Ausführungszeitpunkt(e) des Tasks durch das Scheduling System angezeigt. Sind noch keine Zeitpläne und -punkte angelegt, so ist dieses Feld leer. Sobald ein Zeitplan angelegt wird und Zeiten definiert wurden, wird der nächstmögliche Termin nach diesem Zeitplan angezeigt. Ist kein Zeitpunkt möglich, weil sich ausschließende Zeiten definiert wurden, so wird ebenfalls kein Wert angezeigt.
Wenn Kalendereinträge erstellt werden, wird hier nicht nur der nächste Ausführungszeitpunkt, sondern die nächste fünf Ausführungszeitpunkte, so weit diese innerhalb des Horizonts liegen, gezeigt.
Das Datumsformat ist identisch mit dem Feld *Start*.

18.4.1 Teilbereich Sub Schedule Details

Dieser Teilbereich zeigt alle "Detail" Informationen zum ersten Sub Schedule oder einen mittels des *Edit* Buttons gewählten anderen Sub Schedules an. Die Sub Schedule Details beginnen auf der Maske mit einer grauen Zeile, welche den Namen des aktuell ausgewählten Sub Schedules anzeigt, gefolgt von folgenden Feldern:

Error Code Im Feld *Error Code* wird, falls ein Fehler bei der Ausführung des Jobs im Time Scheduling aufgetreten ist, der übermittelte Fehlercode angezeigt. Ist kein Fehler aufgetreten, bleibt das Feld leer.

Error Message Im Feld *Error Message* wird, falls ein Fehler bei der Ausführung des Jobs im Time Scheduling aufgetreten ist, die übermittelte Fehlermeldung angezeigt. Ist kein Fehler aufgetreten, bleibt das Feld leer.

Liste Intervals Wird hier kein Intervall eingetragen (leer Liste) so wird, falls der Sub Schedule als Active markiert ist, genau einmal zum Startzeitpunkt des Sub Schedules ein Job submittet.
Eine nicht leere Liste muss genau ein "driving" Intervall vom Typ "Time Of Day", "Repeat" oder "Calendar (Driver)" enthalten. Das "driving" Intervall gibt den Takt an, zu dem potentiell ein Submit stattfindet. Die anderen "filtering" Intervalle bestimmen durch Ihre Filterwirkung, welche durch das "driving" Intervall erzeugten Takte tatsächlich zu einem Submit führen.

Type Das Feld *Type* gibt die Art des Intervalls an. Abhängig vom Typ des Intervalls wird in der Spalte *Setup* eine andere Maske angezeigt.

Setup Hier wird abhängig vom Type des Intervalls eine Maske zur Konfiguration des Intervalls angezeigt.

Selection Bei Intervallen, welche eine Selektion ermöglichen (z.B.: ”Day Of Week”) kann in dieser Spalte ausgewählt werden ob die Selektion ”NORMAL” oder ”INVERSE” erfolgen soll.

Es gibt folgende Intervall Typen:

- Repeat

  Hierbei handelt es sich um ein ”driving” Intervall, das den zeitgesteuerten Start des Tasks in regelmäßigen Abständen vorsieht. Die Anzahl der Minuten, kann im Feld *Setup* eingeben werden. Das sieht folgendermaßen aus:

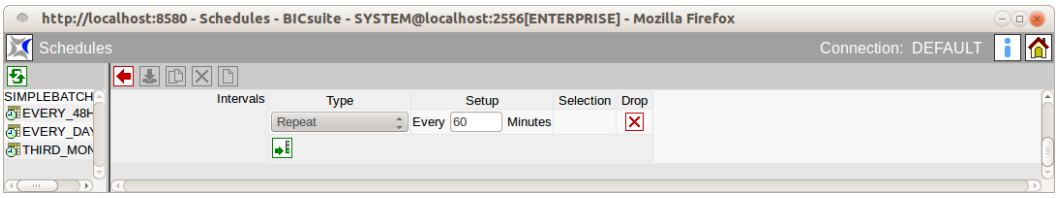


Abbildung 18.4: Repeat Driver

  Im oberen Beispiel, wird der Job alle 60 Minuten gestartet.

- Time of Day

  Mittels dieses ”driving” Intervall Typs kann der Job zu einer vorgegebenen Tageszeit starten. Die Tageszeit wird im Feld *Setup* mittels der Auswahl der Stunden (24H) und der Minuten eingetragen werden. Die Option sieht wie

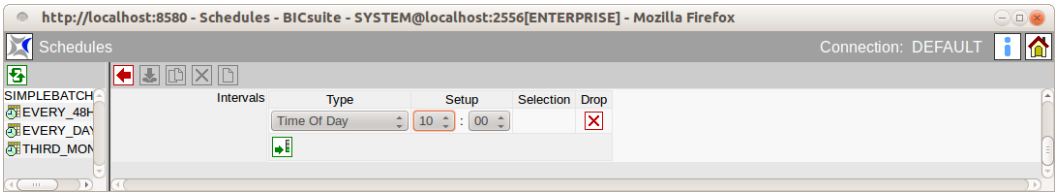


Abbildung 18.5: Time Of Day Driver

  im Bild 18.5 aus.

  Im oberen Beispiel wird der Job jeden Tag um 10:00 Uhr morgens gestartet.

  Es können mehrere ”Time Of Day” Intervalle angelegt werden. Damit ist es dann leicht möglich mehrere Ausführungszeitpunkte an einem Tag zu definieren.

- Range of Day

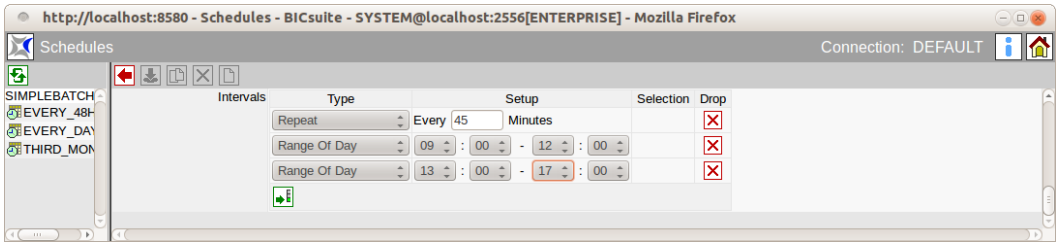


Abbildung 18.6: Range of Day Filter

Dieser ”filtering” Intervall Typ, ist nur in Kombination mit den ”driving” Intervall Typen ”Repeat” und ”Calendar (Driver)” erlaubt. Er erlaubt die Beschränkung auf einen oder mehrere Zeitbereiche innerhalb eines Tages.

Es können mehrere ”Range Of Day” Intervalle angelegt werden. Damit ist es dann leicht möglich mehrere Ausführungszeiträume an einem Tag zu definieren.

- Day of Week

  Mittels des ”filtering” Intervall Typs ”Day of Week” kann ausgewählt werden, an welchen Wochentagen der Job ausgeführt werden soll. Dies wird im Feld *Setup* durch Auswahl der einzelnen Wochentage festgelegt. Die Option

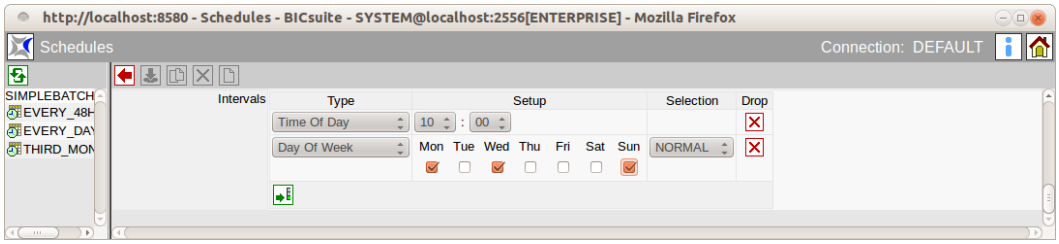


Abbildung 18.7: Day of Week Filter

sieht wie im Bild 18.7 aus.

Im oberen Beispiel wird der Job jeden Montag, Mittwoch und Sonntag ausgeführt.

- Day of Month

  Mittels des ”filtering” Intervall Typs ”Day of Month” Kann ausgewählt werden, an welchen Tagen des Monats der Job ausgeführt werden soll. Diese werden im Feld *Setup* durch Auswahl der einzelnen Tage im Monat festgelegt. Die Option sieht wie im Bild 18.8 aus.

  Im oberen Beispiel wird der Job am 9., 13., 19., 26. und am vorletzten Tag jeden Monats ausgeführt.

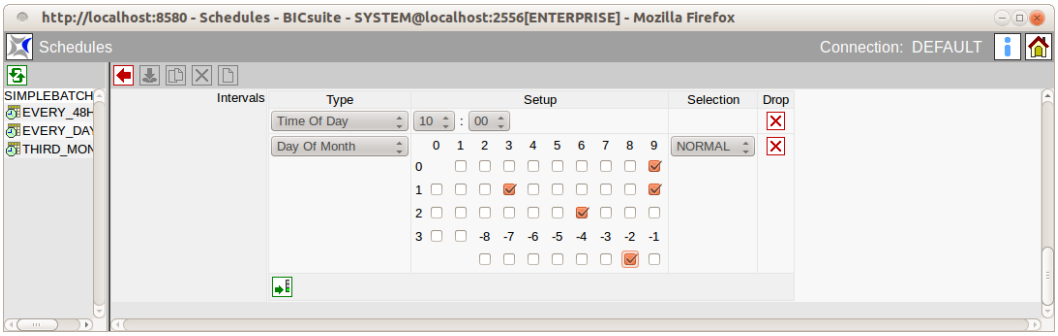


Abbildung 18.8: Day of Month Filter

- Week of Month

  Mittels des ”filtering” Intervall Typs ”Week of Month” kann ausgewählt werden, in welcher Woche des Monats der Job ausgeführt werden soll. Diese werden im Feld *Setup* durch Auswahl der einzelnen Wochen im Monat festge-

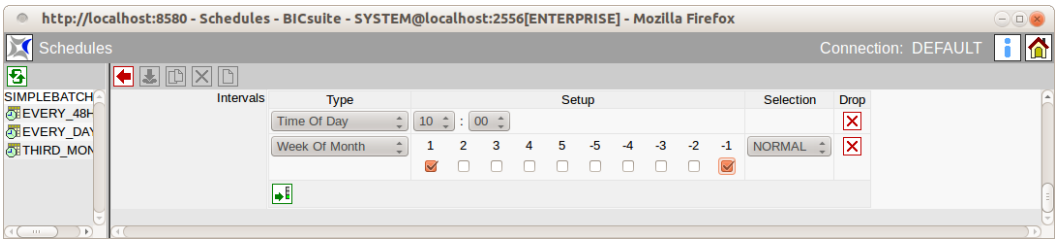


Abbildung 18.9: Week of Month Filter

  legt. Die Option sieht wie im Bild 18.9 aus.

  Im oberen Beispiel wird der Job in der ersten und letzten Woche eines jeden Monats ausgeführt. Dabei gilt, dass die erste Woche als die ersten 7 Tage des Monats und die letzte Woche, als die letzten 7 Tage des Monats definiert ist.

- ISO Week of Month

  Mittels des ”filtering” Intervall Typs ”ISO Week of Month” kann ausgewählt werden, in welcher ISO Woche des Monats der Job ausgeführt werden soll. Diese werden im Feld *Setup* durch Auswahl der einzelnen Wochen im Monat festgelegt. Die Option sieht wie im Bild 18.10 aus.

  Im oberen Beispiel wird der Job am Montag der ersten ISO Woche eines jeden Monats ausgeführt. ISO Wochen beginnen immer Montags. Wochen werden einem Monat zugeordnet wenn mindestens 4 Tage der Woche in diesem Monat liegen. Ist zum Beispiel der erste Wochentag eines Monats ein Mittwoch, so beginnt die erste ISO Woche des Monats schon im Vormonat.

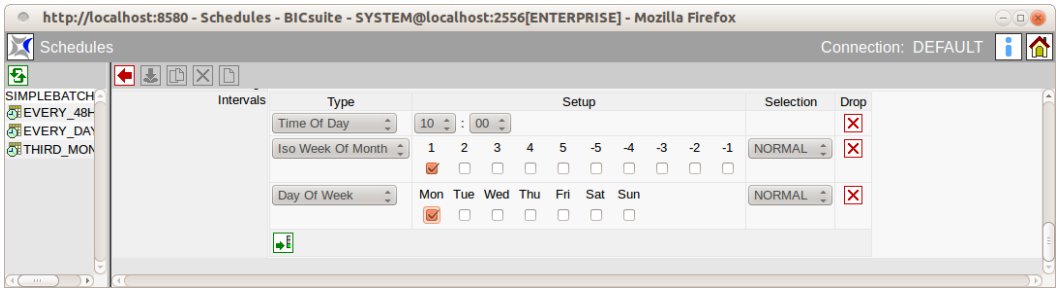


Abbildung 18.10: ISO Week of Month Filter

- ISO Week of Year

  Mittels des ”filtering” Intervall Typs ”ISO Week of Year” kann ausgewählt werden, in welcher ISO Kalenderwoche Woche des Jahres der Job ausgeführt werden soll. Diese werden im Feld *Setup* durch Auswahl der einzelnen Wo-

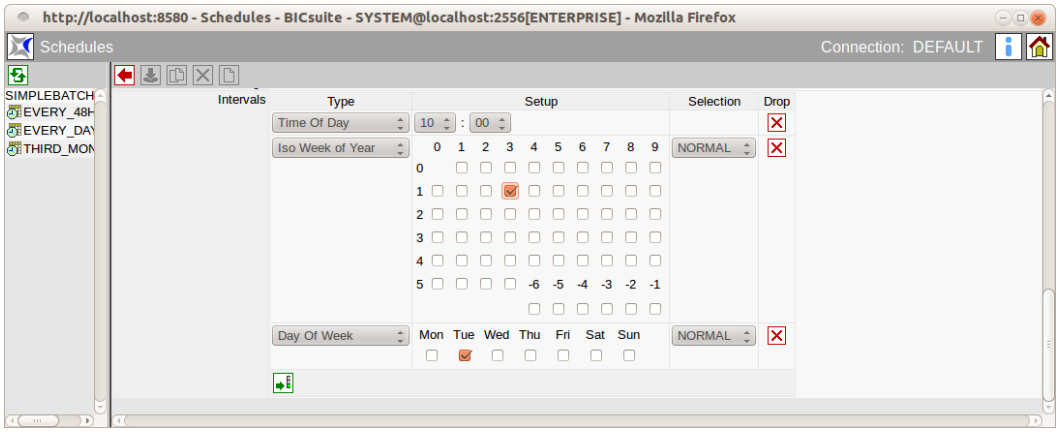


Abbildung 18.11: Week of Year Filter

chen im Jahr festgelegt. Die Option sieht im Bild 18.11 aus.

Im oberen Beispiel wird der Job am Dienstag der 31. ISO Kalenderwoche ausgeführt. ISO Wochen beginnen immer Montags. Wochen werden einem Jahr zugeordnet wenn mindestens 4 Tage der Woche in diesem Jahr liegen. Ist zum Beispiel der erste Wochentag eines Jahres ein Mittwoch, so beginnt die erste ISO Woche des Jahres schon im Vorjahr.

- Month of Year

  Mittels des ”filtering” Intervall Typs ”Month of Year” kann ausgewählt werden, in welchem Monat des Jahres der Job ausgeführt werden soll. Diese werden im Feld *Setup* durch Auswahl der einzelnen Monate im Jahr festgelegt.

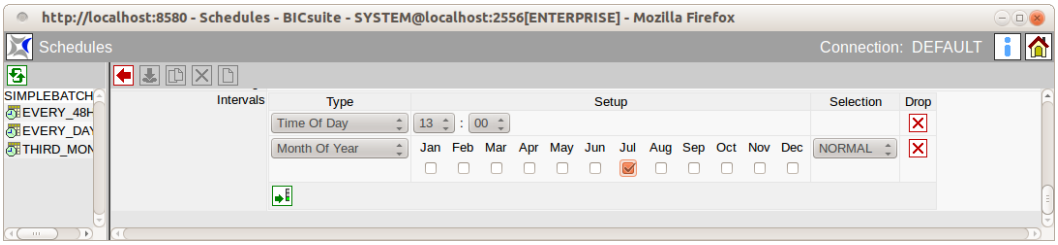


Abbildung 18.12: Month of Year Filter

Die Option sieht wie im Bild 18.12 aus.

Im oberen Beispiel wird der Job jeden Tag im Juli um 13:00 Uhr ausgeführt.

- Calendar (Driver)

  Mittels des ”driving” Intervall Typs ”Calendar (Driver)” können die Ausführungszeitpunkte aus einem vordefinierten Kalender entnommen werden. Diese Option steht nur zur Verfügung, wenn der Administrator solche Kalender angelegt und in BICsuite!web bekannt gemacht hat.

  Im Feld *Setup* kann dann der Kalender ausgewählt, und durch die Option ”with select on” bestimmt werden auf welche Einheit (DAY, WEEK, MONTH, YEAR) sich die folgende Auswahl bezieht.

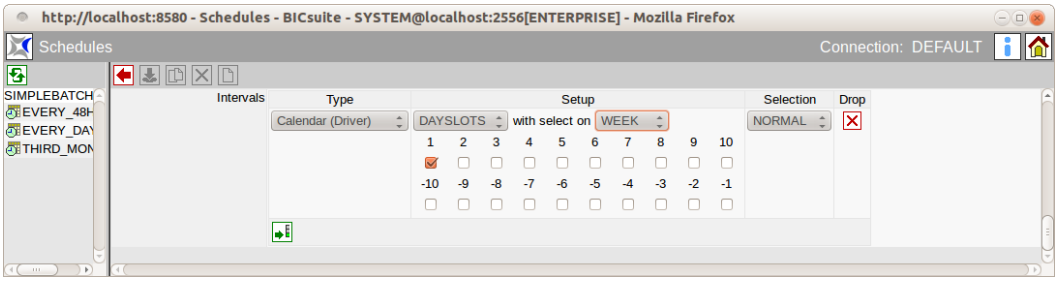


Abbildung 18.13: Calendar Driver

Im oberen Beispiel wird aus einem Kalender mit Namen ”DAYSLOTS” der jeweils erste Eintrag jeder Kalenderwoche ausgewählt.

- Calendar (Filter)

  Mittels des ”filtering” Intervall Typs ”Calendar (Filter)” können die Ausführungszeitpunkte mittels eines vordefinierten Kalender eingeschränkt werden. Diese Option steht nur zur Verfügung, wenn der Administrator solche Kalender angelegt und in BICsuite!web bekannt gemacht hat.

  Im Feld *Setup* kann dann der Kalender ausgewählt, und durch die Option

”with select on” bestimmt werden auf welche Einheit (DAY, WEEK, MONTH, YEAR) sich die folgende Auswahl bezieht.

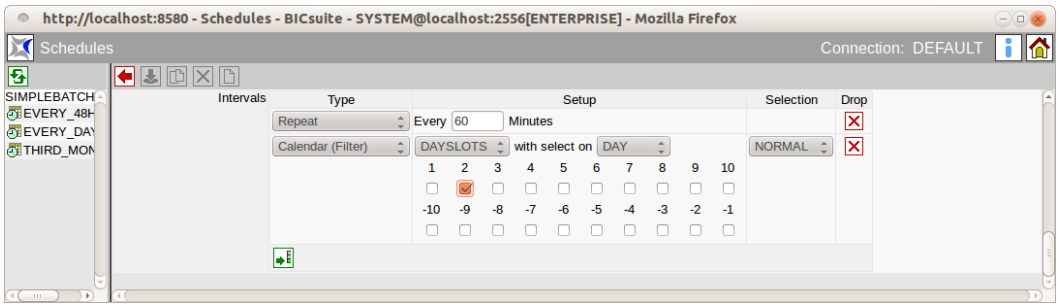


Abbildung 18.14: Calendar Filter

Im oberen Beispiel wird alle 60 Minuten ein Submit ausgeführt falls sich dieser im Kalender ”DAYSLOTS” im zweiten Block eines jeden Tages befindet.

Setup Das Feld *Setup* gibt die genauere Definition des Intervalls an. Es ist je nach Intervalltyp unterschiedlich.

19 Submit Batches und Jobs

19.1 Bild

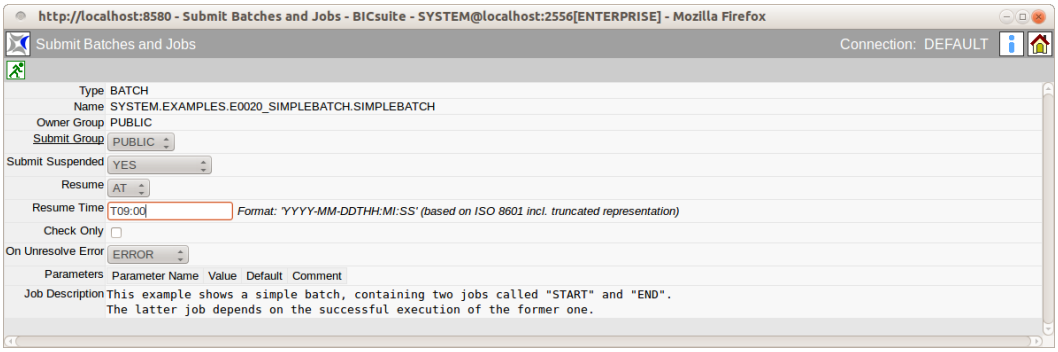


Abbildung 19.1: Submit Batches und Jobs

19.2 Konzept

19.2.1 Kurzbeschreibung

Der Dialog ”Submit Batches and Jobs” dient zum manuellen Starten von, im Dialog Batches and Jobs, definierten Master Jobs. Durch den Submit wird dem BICsuite Server mitgeteilt, das dieser Master Job aktiviert werden soll.

19.2.2 Ausführliche Beschreibung

Die manuelle Freigabe von Master Jobs, ist eine von zwei Methoden um einen Job zu starten. Die zweite Möglichkeit besteht durch den Eintrag des Jobs im Time Scheduling.
Beim manuellen Start können darüber hinaus noch bestimmte Parameter gesetzt und bestimmt werden ob es sich um einen echten Start oder nur einen Check auf Startfähigkeit handelt.
Weiter kann bestimmt werden ob der Master Job gleich Suspended werden soll.

19.3 Navigator

Die Navigation zeigt alle verfügbaren Master Jobs in einer Ordnerhierarchie an. Sie entspricht dabei dem Navigationsbildschirm aus dem Batches und Jobs Dialog. Als Unterschied werden hier allerdings nur Objekte angezeigt, welche das Flag "Submit as Master allowed" gesetzt haben.

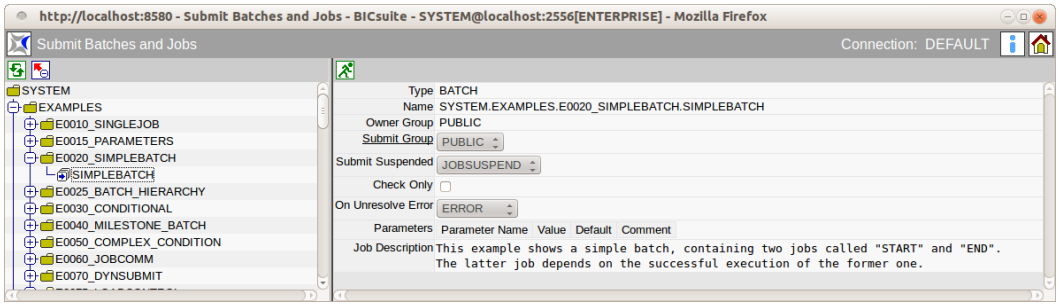


Abbildung 19.2: Submit Navigation

19.4 Editor

Mit dem Editor Frame können Startoptionen und Parameter übergeben und ein Master Job submitted werden.

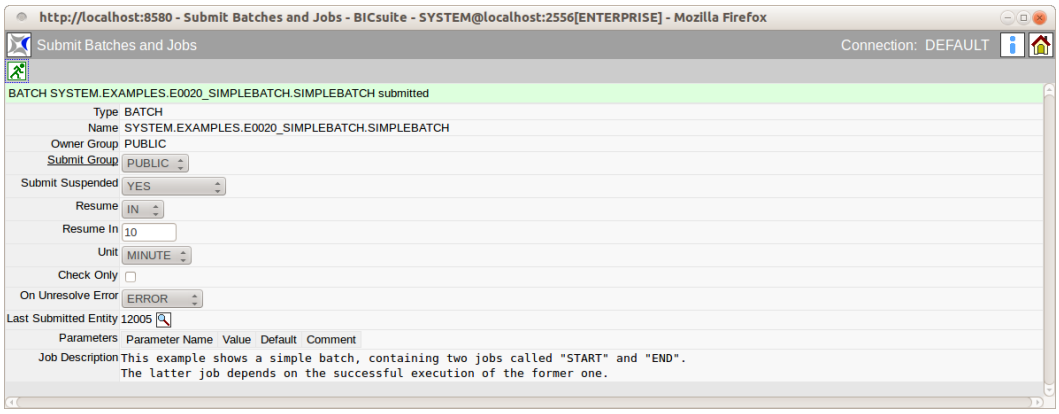


Abbildung 19.3: Submit Editor

Folgender Button ist auf dieser Maske aktiv.

 Submit

Mit dem Submit Button wird der Job gestartet (Flag "Check Only" nicht gesetzt), oder es wird geprüft, ob der Master Job startfähig ist (Flag "Check-Only" gesetzt).

Die obigen Felder haben folgende Bedeutung:

Type Der Typ des Master Jobs. Dieser kann entweder ein Job oder Batch sein.

Name Das ist der vollständige Name (mit Pfad) des Master Jobs.

Owner Group Im Feld *Owner Group* ist der Eigentümer des Jobs definiert.

Submit Group Die *Submit Group* bestimmt die Eigentümergruppe des Jobs, der Submitted wird.

Submit Suspended Dieses Feld gibt an ob der Job gleich nach dem Start in den Suspend Status gehen soll oder nicht.
Die folgenden Eingabefelder *Resume*, *Resume Time*, *Resume In*, *Unit* werden nur angezeigt falls dieses Feld auf 'YES' gesetzt wird.
Es gibt folgende Optionen:

1. YES

   Das Child wird beim Submit als Suspended angelegt und muss durch eine explizite Freigabe gestartet werden.

2. NO

   Das Child wird beim Submit nicht Suspended und kann sofort starten.

3. JOBSUSPEND

   Ob eine Verzögerung stattfindet oder nicht, hängt vom Feld *Suspend* des zu submittenden Jobs ab. Das heißt, je nachdem wie die Einstellung im Job definiert wurde, wird diese übernommen.

Resume Hier kann ausgewählt werden, ob ein automatischer Resume stattfinden soll.
Es gibt folgende Möglichkeiten:

- NO: wählt diese Funktionalität ab und es werden keine anderen Eingabefelder angezeigt.
- AT: wählt einen automatischen Resume zu einem festen Zeitpunkt. Das Eingabefeld *Resume Time* wird angezeigt.
- IN: wählt einen automatischen Resume nach Ablauf einer Zeit. Die Eingabefelder *Resume In* und *Unit* werden angezeigt.

Resume Time Hier wird der gewünschte Resume Zeitpunkt im Format "YYYY-MM-DDTHH:MI:SS" eingegeben.
Das Format orientiert sich an der ISO Norm 8601 und erlaubt auch unvollständige Angaben. Die Eingabe von 'T09:00' wird den Job um 09:00 Uhr resumen (ausgehend von der aktuellen Zeit).

Resume In Hier wird angegeben, wie viele Zeiteinheiten (siehe *Unit*) bis zum Resume gewartet werden soll.

Unit Hier wird eingegeben ob es sich bei der Eingabe im *Resume In* um Minuten (MINUTE), Stunden (HOUR), oder Tage (DAY) handeln soll.

On Unresolve Error Im Feld *On Unresolve Error* wird beim Submitten bestimmt, wie man im Falle von nicht aufgelösten Abhängigkeiten verfährt.
Es gibt folgende Möglichkeiten:

- Error: Die Abhängigkeiten die nicht aufgelöst werden können, werden als Fehler betrachtet. Die fehlenden Abhängigkeiten führen zum Submit-Abbruch.
- Suspend: Beim Suspend erfolgt der Submit als wären diese Abhängigkeiten nicht definiert. Allerdings wird der Job suspended submitted.
- Ignore: Beim Ignore wird submittet, als wären diese Abhängigkeiten nicht definiert.

Check Only Dieses Flag gibt an ob der Job submitted wird oder ob nur eine Prüfung der Startfähigkeit stattfindet.

Liste Parameters Benötigt der Job Parameter, müssen die Parameterwerte vor dem Submit in dieser Liste eingetragen werden. Mehr zum Thema Parameter finden Sie im Kapitel 14.5.10.

Job Description Die Beschreibung des Jobs aus der Job Definition.

20 Bookmarks

20.1 Bild

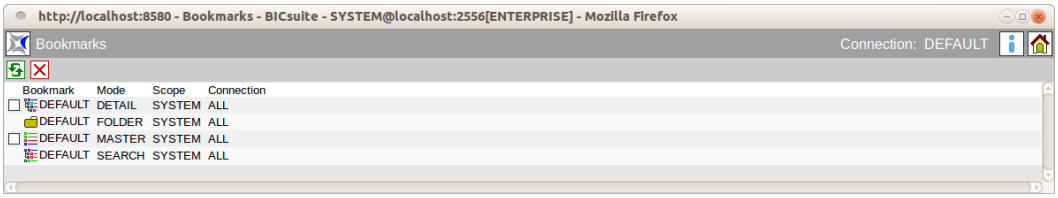


Abbildung 20.1: Bookmarks

20.2 Konzept

20.2.1 Kurzbeschreibung

Im Dialog ”Bookmark” ist es möglich gespeicherte Abfragen aus den Dialogen ”Running Master Jobs” und ”Search Running Jobs” abzurufen. Hiermit sind immer wiederkehrende Suchabfragen und Übersichtsfenster leicht zu speichern und wieder neu aufzurufen.

20.2.2 Ausführliche Beschreibung

Bookmarks können für den aktuellen Benutzer oder systemweit zur Verfügung gestellt und geändert werden. Darüber hinaus ist es möglich, bestimmte Bookmarks immer sofort nach dem Anmelden ins System zur Verfügung zu haben. Diese werden dementsprechend automatisch gestartet.
Die Liste der Bookmarks enthält folgende Felder:

Bookmark Hier wird der Name des Bookmarks angezeigt. Durch Anwahl des Namens erscheint ein neues Fenster mit der als Bookmark gespeicherten Anfrage.

Mode Im Feld *Mode* steht die Art eines Bookmarks. Es gibt folgende Arten:

1. FOLDER

   Hierbei handelt es sich um einen Folder Bookmark.

2. MASTER

   Hierbei handelt es sich um einen Bookmark, welcher von der Suchmaske die im Kapitel 21 beschrieben wird, gespeichert wurde. Mittels dieses Bookmarks sind nur Master Jobs sichtbar.

3. SEARCH

   Der Mode Search beschreibt einen Bookmark, welcher von der Suchmaske die im Kapitel 22 beschrieben wird, gespeichert wurde. Hiermit sind alle Jobs auffindbar.

4. DETAIL

   Hiermit wird ein Bookmark beschrieben, der die ”Detail” Ansicht (mehr hierzu im Kapitel 21) beschreibt. Dieses CHILD hängt immer mit einem Bookmark des Modes MASTER zusammen (ist das Child der ”Master” Ansicht).

   Welcher Bookmark gewählt wird, wird wie folgt ermittelt:

   Definiert der Master Job oder Batch einen Parameter DETAIL_BOOKMARK so wird dieser verwendet. Ist dies nicht der Fall, wird das Feld ’Detail Bookmark’ des Master Bookmarks ausgewertet. Ist dies ebenso nicht gesetzt so wird der Detail Bookmark ’DEFAULT’ verwendet.

Scope Mit dem Scope Parameter wird angegeben ob der Bookmark systemweit oder nur für den aktuellen Benutzer sichtbar ist. Es gibt folgende Optionen:

1. SYSTEM

   Ein Bookmark mit dem Scope SYSTEM ist für alle Benutzer im gesamten System sichtbar. Er wird in der Bookmark-Liste von allen Benutzern angezeigt und darf von allen verwendet werden. Werden Änderungen an einem Bookmark vom Scope SYSTEM durchgeführt, gelten diese systemweit für alle Benutzer.

   Bookmarks vom Typ SYSTEM dürfen allerdings nur von Benutzern mit Web-GUI-Administrationsrechte (siehe Web-Users) angelegt werden. Ein normaler Benutzer darf zwar SYSTEM Bookmarks ändern, der Bookmark wird aber anschließend als USER Bookmark gespeichert. Das heißt, die lokale Änderung überschreibt den Bookmark nur für den aktuellen Benutzer, nicht für andere. Löscht man diesen Bookmark, erscheint wieder der Original SYSTEM Bookmark.

2. USER

   Hat der Scope den Wert User, dann ist dieser Bookmark von dem aktuellen User angelegt worden und nur für diesen verfügbar.

Connection Das Feld *Connection* informiert über die Gültigkeit des Bookmarks bei Verwendung mehrerer Server Verbindungen. Folgende Optionen existieren:

1. CURRENT

   Der Bookmark gilt nur für die aktuelle Connection. Wird das Bookmark Fenster für eine andere Connection geöffnet ist dieser Bookmark nicht sichtbar.

2. ALL

   Der Bookmark ist für alle Server Connections gültig.

20.3 Navigation

Der Bookmark-Dialog besteht nur aus einem Navigationsfenster. Wird aus dem Navigationsfenster ein Eintrag gewählt, wird dieser in einem neuen Fenster angezeigt.

21 Running Master Jobs

21.1 Bild

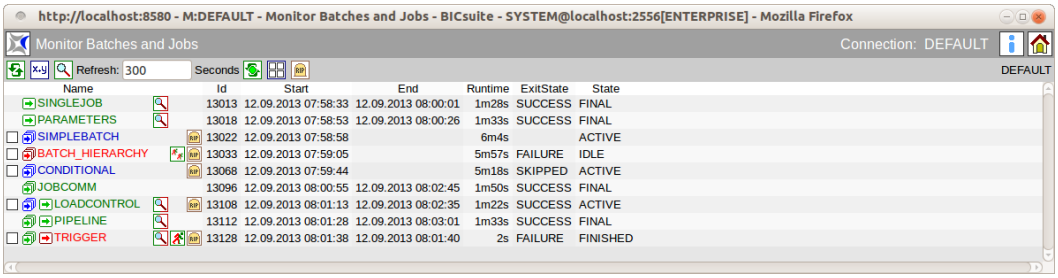


Abbildung 21.1: Running Master Jobs

21.2 Konzept

21.2.1 Kurzbeschreibung

Der Dialog Running Master Jobs dient zur Anzeige der aktuell laufenden oder historisch gelaufenen (je nach Einstellung) Master Jobs des Systems.

21.2.2 Ausführliche Beschreibung

Der Dialog *Running Master Jobs* dient zur Anzeige der aktuell laufenden Jobs (Monitoring) sowie die Betreuung der aktiven Jobs (Operating). Die Anzeige und Filterkriterien des Running Master Job Dialoges können über den Bookmark (Default, Master, System) angepasst werden. Dieser Bookmark beschreibt die Anzeige des kompletten Dialoges.

21.3 Master Navigator

Die Navigation nimmt den ganzen Raum des Dialoges ein. Hier erscheint eine Liste aller aktuell laufenden Master-submittable Jobs und Batches.

 Find

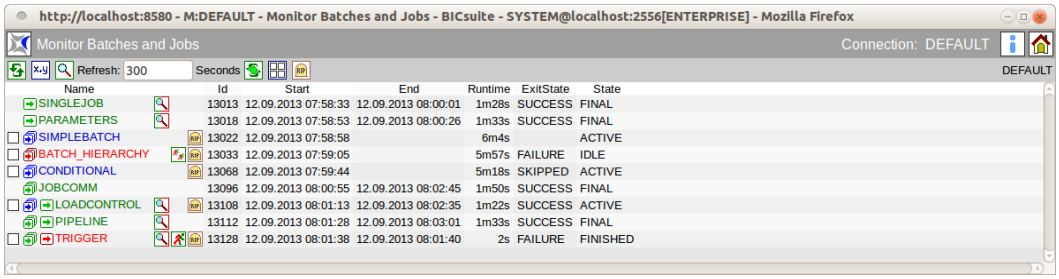


Abbildung 21.2: Running Master Jobs Navigator

Betätigt man den *Find* Button, so kommt man zur Seite für die Eingabe der Suchkriterien sowie Einstellung des Bildschirms.

 Auto-Refresh On/Off

Der *Auto-Refresh On* Button ist ebenfalls ein Schalter und kann in zwei Stellungen betrieben werden. In der Stellung "Auto-Refresh-On", wird in der eingestellten Zeit eine Aktualisierung automatisch durchgeführt.
Im Eingabefeld links vom Button kann die Zeit, die zwischen zwei Refreshes liegt, in Sekunden eingeben werden. Der Default-Wert ist 300 Sekunden. Anhand der Farbe des Buttons kann der Zustand des Auto-Refresh abgelesen werden. Grün = On, Rot = Off.
Die im Dialog erscheinende Liste gibt alle Master Jobs an, die im aktuellen System aktiv sind (default) oder falls über die Suchmaske Filteroptionen eingegeben wurden, die dementsprechend gefilterten Ergebnisse. Folgende Spalten sind in der Liste zu sehen:

Name Hier erscheint je nach Stellung des Buttons Show Jobs, der vollständige Pfad und Name des Master Jobs oder nur der Name des Master Jobs.
Der Name ist anklickbar und führt zur Detailmaske falls es sich um einen Master Submittable Batch mit Children handelt, ansonsten führt das Klicken zur Detailmaske für Jobs.
Vor dem Namen können Icons angezeigt werden, welche bestimmte States und Situation beschreiben. Es gibt folgende Icons:



Beim angezeigten Objekt handelt es sich um einen Job der aktuell läuft oder laufen kann. Die Icon-Farbe ist blau.



Beim angezeigten Objekt handelt es sich um einen Job der wartet. Die Icon-Farbe ist lila.

- Job Icon ohne Punkt: Der Job befindet sich im Dependency Wait.

- Job Icon mit einem Punkt: Der Job befindet sich im Synchronize Wait.
- Job Icon mit zwei Punkten: Der Job befindet sich im Resource Wait.



Beim angezeigten Objekt handelt es sich um einen Job der vom Time Scheduling in der Zukunft submitted werden wird.



Beim angezeigten Objekt handelt es sich um einen Job der vom Time Scheduling in der Zukunft "suspended" submitted werden wird.



Beim angezeigten Objekt handelt es sich um einen Job der gecancelt wurde. Der Zustand dieses Jobs kann sich nicht mehr ändern. Die Icon-Farbe ist braun.



Beim angezeigten Objekt handelt es sich um einen Job der Final ist. Der Zustand dieses Jobs kann sich nicht mehr ändern. Die Icon-Farbe ist grün.



Beim angezeigten Objekt handelt es sich um einen Job der Finished und Restartable, Unreachable oder Error ist. Die Icon-Farbe ist rot.



Beim angezeigten Objekt handelt es sich um einen Batch der im Moment läuft. Die Icon-Farbe ist blau.



Beim angezeigten Objekt handelt es sich um einen Batch der im Moment wartet. Die Icon-Farbe ist lila.



Beim angezeigten Objekt handelt es sich um einen Batch der vom Time Scheduling in der Zukunft submitted werden wird.



Beim angezeigten Objekt handelt es sich um einen Batch der vom Time Scheduling in der Zukunft "suspended" submitted werden wird.



Beim angezeigten Objekt handelt es sich um einen Batch der final ist. Der Zustand des Batches kann sich nicht mehr ändern. Die Icon-Farbe ist grün.



Beim angezeigten Objekt handelt es sich um einen Batch der gecancelt wurde. Der Zustand dieses Batches kann sich nicht mehr ändern. Die Icon-Farbe ist braun.



Beim angezeigten Objekt handelt es sich um einen Batch der Unreachable ist, oder von dem mindestens einer der Kinder Finished und restartable, Unreachable oder Error ist. Die Icon-Farbe ist rot.



Beim angezeigten Objekt handelt es sich um einen Milestone der wartet. Die Icon-Farbe ist lila.



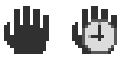
Beim angezeigten Objekt handelt es sich um einen Milestone der gecancelt wurde. Die Icon-Farbe ist braun.



Beim angezeigten Objekt handelt es sich um einen Milestone der Final ist. Die Icon-Farbe ist grün.



Beim angezeigten Objekt handelt es sich um einen Milestone der Unreachable ist, oder von dem mindestens einer der Kinder Finished und restartable, Unreachable oder Error ist. Die Icon-Farbe ist rot.



Diese Icons teilen dem Benutzer mit, dass dieses Objekt Suspended wurde. Das Icon mit der Uhr zeigt an, dass für das Objekt ein automatischer Resume eingestellt wurde. Die Farben des Namens sind in ihrer Bedeutung analog zu den farblich identischen Buttons.



Der *Cancel Run* Button erscheint, falls der Job auf Ressourcen oder Abhängigkeiten wartet oder noch auf einem Jobserver zugeteilt wurde. Durch Drücken des *Cancel Run* Buttons kann der geplante Joblauf abgebrochen werden. Der Job bekommt den State "Cancelled". Es ist nun kein Rerun und kein Resume möglich. Der Job bekommt auch keinen Exit State. Jobs, welche auf einen gecancelten Job warten, müssen diesen Job ignorieren oder ebenfalls gecancelt werden.



Mittels dieses Buttons kann ein neues Fenster aufgemacht werden, welches den Inhalt des Logfiles darstellt. Sollte dies nicht möglich sein, wenden Sie sich bitte an Ihren Systemadministrator.

 Rerun

Der *Rerun* Button erscheint, falls der aktuelle Job beendet wurde und sich in einen "Restartable" Zustand befindet. Das heißt, er besitzt einen Exit State welcher nicht Final ist. Durch Drücken des *Rerun* Buttons, wird das Rerun Programm aufgerufen (falls definiert, ansonsten das Run-Kommando). Hiermit ist ein erneutes Starten eines fehlerhaften Jobs, nach der Problembehebung möglich.

 Rerun Children

Der *Rerun Children* Button wird angezeigt, wenn sich Children des aktuellen Jobs in einem "Restartable" Zustand befinden. Das heißt, sie besitzen einen Restartable Exit State. Durch Drücken des *Rerun Children* Buttons, wird das Rerun für alle Children gestartet, welche Restartable sind. Hiermit ist ein erneutes Starten fehlerhafter Children nach einer Problembehebung möglich.



Diese Icons werden angezeigt wenn eine Warnung ausgegeben wurde. Eine Warnung wird z. B. abgegeben wenn ein Trigger ausgelöst wurde, der Submit jedoch fehlgeschlagen ist. Im Falle einer Warnung können genauere Informationen dem Audit entnommen werden.
Das zweite, hellere Icon wird gezeigt, wenn bei einer der Kinder eine Warnung ausgegeben wurde.

Id Die *ID* ist der eindeutige Identifier der aktuellen "Master Job"-Laufzeitinstanz im System.

Start Beim Feld *Start* handelt es sich um den Zeitpunkt des Submits oder des Starts.

End Beim Feld *End* handelt es sich um den Zeitpunkt der Beendigung des Jobs. Wurde der Job noch nicht beendet, bleibt der Eintrag leer.

Runtime Die Runtime gibt die aktuelle Laufzeit des Jobs in Tage (d), in Stunden (h), in Minuten (m) und Sekunden (s) an. Läuft der Job noch, wird die Laufzeit bis zum letzten Refresh des Dialoges angezeigt und bei jedem neuen Refresh aktualisiert.

Exit State Wurde der Job beendet, wird der Exit State den der Job bei der Beendigung bekommen hatte angezeigt. Ist der Job noch nicht beendet, bleibt das Feld leer.

State Der Job State ist der aktuelle Laufzeitstatus des Jobs. Das Laufzeitsystem des BICsuite Serversystems vergibt und ändert diese States, wenn der Job (das

heißt, die Laufzeitinstanz des Jobs) seinen Lebensweg durch das BICsuite System durchläuft. Der Job kann folgende States haben:

1. Submitted

   Der Job wurde manuell, durch einen Parent-Prozess oder über das Time Scheduling submitted und soll ausgeführt werden. Dies ist der initiale Status eines Jobs und ist normalerweise nicht sichtbar.

2. Dependency Wait

   Der Job wartet auf notwendige Abhängigkeiten die erfüllt werden müssen.

3. Synchronize Wait

   Der Job wartet auf benötigte "Synchronizing Resources".

4. Resource Wait

   Der Job wartet auf die Bereitstellung ausreichender System Resources.

5. Unreachable

   Der Job kann nicht ausgeführt werden, da eine oder mehrere Abhängigkeiten (Dependencies) nicht erfüllt sind. Diese Situation kann durch das Ignorieren von Abhängigkeiten behoben werden.

6. Cancelled

   Der Job wurde manuell gecanceled und wird nicht mehr ausgeführt.

7. Error

   Durch einen Definitionsfehler kann der Job nicht ausgeführt werden. Ein Beispiel hierfür ist etwa eine benötigte Resource, die von keinem Jobserver zur Verfügung gestellt wird. In so einem Fall kann der Job restarted werden, nachdem die Fehlersituation behoben wurde.

   Es kommt auch häufig vor, dass bei der Eingabe des Run Programs ein Fehler gemacht wird. Wenn daraufhin der Jobserver nicht in der Lage ist den Prozess zu starten, wird der Job in den Zustand Error versetzt. Auch in dem Fall kann der betreffende Job restarted werden.

   Falls der Submit eines Masters über das Time Scheduling fehl schlägt, wird der Master zur Kennzeichnung dieses Fehlers im Status Error in das System eingestellt. Damit ist der Fehler für den Verantwortlichen sichtbar. Der Job oder Batch ist jedoch nicht restartable und muss, nach der Reparatur, manuell submitted werden.

8. Runnable

   Der Job kann von einem Jobserver gestartet werden. Verbleibt ein Job längere Zeit in diesem Zustand, sollte der Jobserver überprüft werden.

9. Starting

   Der Job wurde einem Jobserver zum Starten übergeben.

10. Started

    Der Job wurde einem Jobserver zum Starten übergeben. Der Jobserver hat den Auftrag Quittiert.

11. Running

    Das Run Command wurde vom Jobserver gestartet.

12. To Kill

    Der Benutzer hat den BICsuite Server beauftragt das zu dem Job gehörige Kill Program aufzuführen. Dass Kill Programm kann mittels des *Cancel-Run* Buttons ausgeführt werden.

13. Killed

    Das zum Job gehörige Kill Program wurde ausgeführt.

14. Broken Active

    Der Jobserver hat seine Verbindung zum Run Program verloren. Der Prozess läuft zwar noch, aber das Ergebnis des Jobs (Exit-Code) kann nicht mehr an den Jobserver und dementsprechend an den BICsuite Server zurückgegeben werden.

15. Broken Finished

    Hat ein Job einen State "Broken Active" und beendet sich, geht er in den State "Broken Finished" über. Das bedeutet, der Prozess läuft nicht mehr. Der Exit Code des Prozesses konnte nicht mehr vom Jobserver ermittelt werden. Eine manuelle Überprüfung der Jobergebnisse und ein manuelles Setzen des Exit State ist nötig. Das Setzen des Exit State erfolgt mittels des Buttons *Set State*.

16. Finished

    Der Job wurde beendet. Der Exit State ist im Feld *Exit State* zu sehen.

17. Final

    Der Job sowie alle seine Children sind beendet und haben ein Final Exit State.

Die Abbildung 21.3 gibt der Statusübergänge des BICsuite Laufzeitsystems wieder.

Variables Nach dem State können, je nach Konfiguration, die dort definierten Variablen in der definierten Reihenfolge und Farbe stehen.

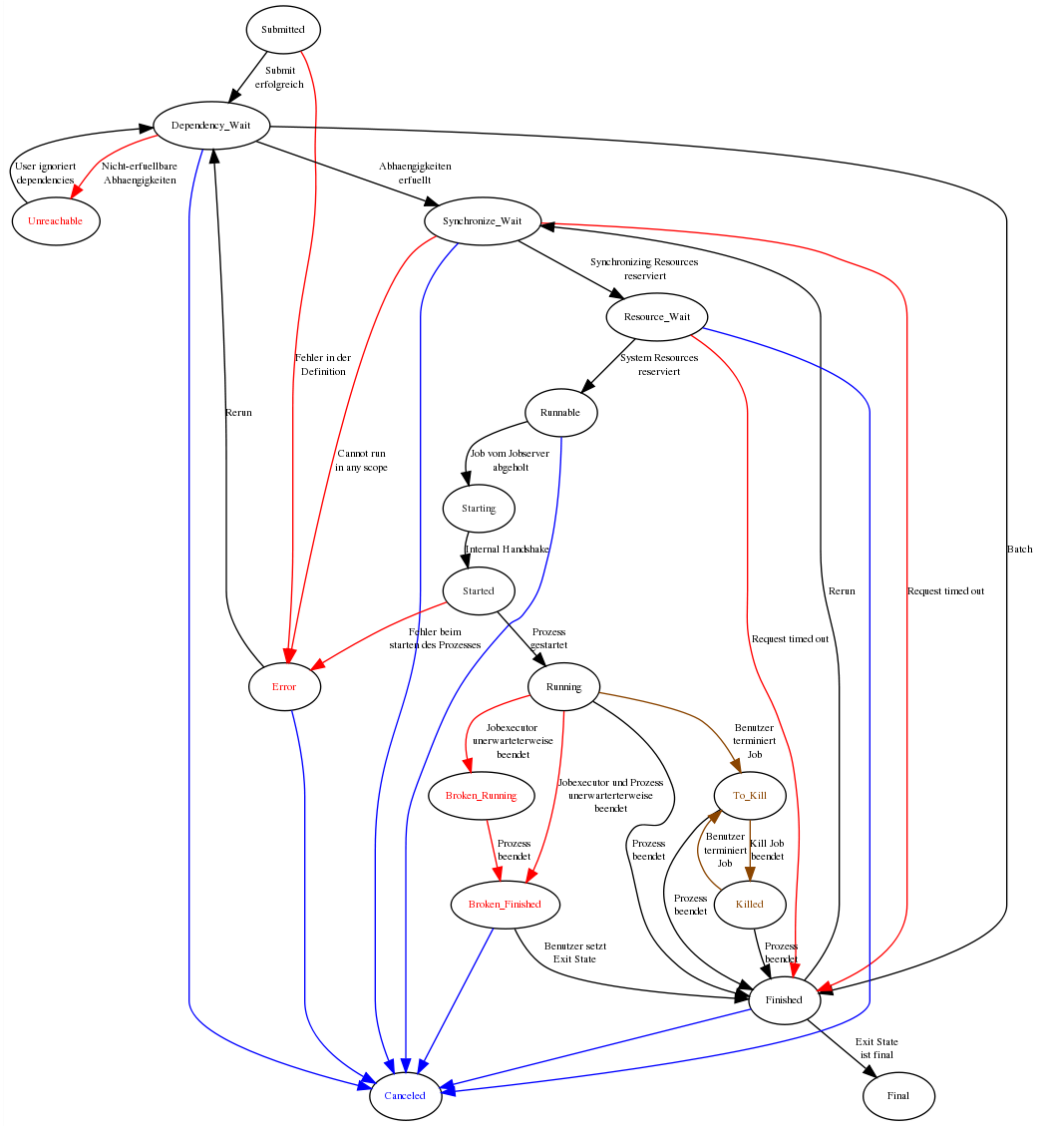


Abbildung 21.3: Statusdiagramm von Batches und Jobs

21.3.1 Master Navigator Query Maske

Durch drücken des *Settings* Button, erscheint die Query Maske. Die Query Maske dient zum Einstellen der Query-Anforderungen für den aktuellen Dialog. Hier können Filterkriterien und Anzeigekriterien editiert werden. Der Dialog sieht folgendermaßen aus:
Durch Veränderung in der Maske und anschließendes Abspeichern unter dem aktuellen oder einem neuen Bookmarknamen (Eingabefeld in der Kopfzeile rechts

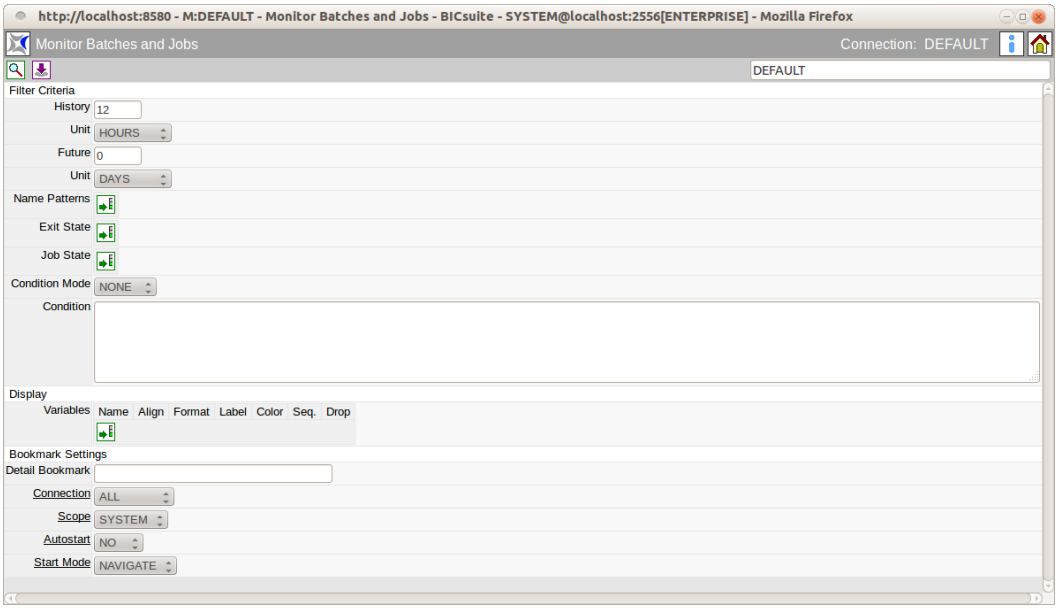


Abbildung 21.4: Running Master Jobs Querymaske

oben), kann die Änderung für eine spätere Wiederverwendung gespeichert werden. Soll die Veränderung nur temporärer Natur sein, so kann sie einfach eingetragen und zum Navigationsbildschirm zurückgesprungen werden. Damit sind die Änderungen nur für die aktuelle Suche gültig. Wird der Dialog anschließend geschlossen, sind die Änderungen verloren.
Die obigen Felder haben folgende Bedeutung:

History/Unit Die maximale Zeit, die seit Ende eines Ablaufes vergangen sein darf, damit er in der Anzeige erscheint. Sollen zum Beispiel alle Jobs die aktuell laufen und alle Jobs, welche in der letzten Stunde beendet wurden angezeigt werden, muss hier der Wert 1 und im Feld *Unit* die Einheit Hours eingestellt werden. Sollen zum Beispiel alle laufenden und alle in der letzten Stunde beendeten Jobs angezeigt werden, muss im Feld *History* der Wert 1 und in der Unit die Einheit Hours eingestellt werden. Es gibt folgende Möglichkeiten:

1. Minutes
2. Hours
3. Days

Future/Unit Über das Feld *Future* wird festgelegt ob und wie weit zukünftig zu submittende Batches bzw. Jobs in der Masterliste angezeigt werden. Unabhängig von dem im Feld *Future* ausgewählten Intervall, kann für Schedules ohne aktiviertem Kalender wird nur der nächste Start angezeigt werden. Für Schedules mit Kalender können nur geplante Starts im Kalenderhorizont des Schedules angezeigt werden. Das Feld *Unit* wird wie bei der History behandelt.

Liste Name Patterns In der Liste der Name Patterns kann eine Menge von Namens Pattern stehen, nach denen die Ergebnismenge gefiltert werden soll. Ein Name Pattern kann der vollständige Name, ein Namensteil oder ein regulärer Ausdruck sein.
Werden in der Liste mehrere Einträge gesucht, wird nach jedem Vorkommen all dieser Einträge gesucht. Diese werden mit Oder verknüpft.
Beispiel:
Es soll nach dem Auftreten des Namens "Worker" gesucht werden.
Folgende Name Patterns sind möglich:

- Vollständig: **Pipeline** (Groß-/Kleinschreibung spielt keine Rolle)
- Teil-String: **eline**
- Regulärer Ausdruck: **P.*line**

Durch die *Hinzufügen* und *Entfernen* Buttons ist die Änderung der Liste der Name Patterns möglich.

Liste Exit State In der Liste können Exit States eingetragen werden, welche die gesuchten Jobs haben müssen, damit sie in der Ergebnisliste angezeigt werden können.
Beispiel:
Es sollen nur Jobs angezeigt werden, deren Exit State "Failure" ist.
Durch Hinzufügen des Exit State "Failure" in die Liste der Exit States, kann nun nach diesen Jobs gesucht werden.
Enthält die Liste keine Einträge, werden die Exit States bei der Suche nicht berücksichtigt.
Durch die Hinzufügen und *Entfernen* Buttons, ist die Änderung der Liste der Exit States möglich.

Liste Job State In der Liste kann nach Job States gesucht werden, die sich in einem der angegebenen Zustände befinden.
Enthält die Liste keine Einträge, werden die Job States bei der Suche nicht berücksichtigt.
Durch die Hinzufügen und *Entfernen* Buttons, ist die Änderung der Liste der Job States möglich.

Condition Mode Der Condition Mode sagt aus, ob und wie die nachfolgende Condition ausgewertet wird.
Es gibt folgende Möglichkeiten:

- None: Die Condition wird nicht berücksichtigt.
- And: Die Suchanfrage und die Condition werden mit "Und" verknüpft.
- Or: Die Suchanfrage und die Condition werden mit "Oder" verknüpft.
- Minus: Die Suchanfrage und die Negation der Condition werden mit "Und" verknüpft.
- Only: Es wird nur die Condition berücksichtigt.

Condition Mit der Condition lässt sich die Suchanfrage verfeinern. Für die genaue Syntax und die verschiedenen Möglichkeiten wird auf die Syntaxdokumentation (List Job) verwiesen.

Liste Variables In der Liste Variables können alle, in der Navigationsliste anzuzeigenden Jobvariablen und Parameter, definiert werden. Hiermit ist eine leichte Übersicht über wichtige Parameter des Jobs möglich. Die Anzeige erfolgt am Ende jeder Zeile in der Navigationsliste.
Bei den Variables müssen folgende Felder eingetragen werden:

Name Der Name der Variablen oder des Parameters muss hier eingetragen werden.

Align Mit dem Feld *Align* kann angeben werden, wie die Anzeige der Variablen in der Liste erfolgen soll. Folgende Möglichkeiten können ausgewählt werden:

- LEFT

  Die Anzeige erfolgt linksbündig.

- CENTER

  Die Anzeige erfolgt zentriert.

- RIGHT

  Die Anzeige erfolgt rechtsbündig.

Format Hiermit kann das Format für die Ausgabe angegeben werden. Es gibt folgende Möglichkeiten:

- KEINE

  Es erfolgt keine Formatierung.

- NUMBER

  Es erfolgt eine numerische Formatierung mit Tausendertrenner und Nachkommastellen (falls nötig).

Label Hiermit kann die Überschrift, welche im List-Kopf für die Variable erscheinen soll, festgelegt werden. Wird kein Label festgelegt, dann wird der Variablenname als Überschrift verwendet.

Color Hiermit kann die Hintergrundfarbe der Spalte für die Variable hinterlegt werden. Im "Drop Down" Menü des Feldes ist eine Vorschau für die jeweilige Farbe zu sehen.

Seq. Über die Buttons *Up* und *Down*, kann die Reihenfolge der Variablen in der Liste eingestellt werden. Die Buttons sind nur zu sehen, wenn mehr als ein Eintrag in der Liste der Variablen erscheint.

Bookmark Settings Die Eingabefelder im Bereich *Bookmark Settings* sind nur beim Speichern eines Bookmarks relevant. Die folgenden Eingabefelder beschreiben Eigenschaften des zu speichernden Bookmarks.

Detail Bookmark Wird im Feld *Detail Bookmark* der Name eines Detail Bookmarks eingegeben, so wird beim Öffnen eines Detail Navigation Fensters aus einem über diesen Bookmark geöffneten Master Navigation Festers der angegebene Detail Bookmark verwendet. Existiert dieser noch nicht, so wird der dieser Bookmarkname in der Detail Navigation in Eckigen Klammern angezeigt. Diese verschwinden beim Anlegen des Detail Bookmarks in der Detail Navigation durch Speichern des Bookmarks. Das Feld *Detail Bookmark* wird nur in den *Bookmark Settings* der Master Navigation angezeigt.

Connection Hier wird festgelegt, für welche Connections, der aktuelle Bookmark gelten soll. Es gibt zwei Ausprägungen:

1. ALL

   Der Bookmark kann für alle Server Connections verwendet werden.

2. CURRENT

   Der Bookmark kann nur für die aktuelle Server Connection verwendet werden.

Scope Hier wird der Gültigkeitsbereich definiert, für den der aktuelle Bookmark gelten soll. Es gibt zwei Ausprägungen:

1. USER

   Der Bookmark ist nur für den aktuellen Benutzer sichtbar. Alle anderen Benutzer sehen diesen Bookmark nicht oder falls es sich um eine Änderung an einen systemweit verfügbaren Bookmark handelt, ist die Änderung nur lokal für diesen Benutzer sichtbar.

2. SYSTEM

   Der Bookmark oder die aktuelle Änderung ist für alle Benutzer sichtbar. Das heißt, alle Benutzer bekommen diesen neuen Bookmark in ihrer Bookmark-Liste zu sehen. Handelt es sich um eine Änderung an einen bestehenden systemweiten Bookmark, trifft die Änderung auf alle Benutzer zu.

Autostart Durch Autostart kann angegeben werden, ob ein Bookmark sofort nach dem Anmelden aktiv werden soll. Das Feld *Autostart* wird nur in den *Bookmark Settings* der Master Navigation angezeigt. Es gibt zwei Ausprägungen:

1. YES

   Der Bookmark wird sofort nach dem Anmelden gestartet. Das heißt, für jeden Bookmark mit Autostart "Yes" erscheint nach dem Anmelden ein Fenster auf dem Bildschirm.

2. NO

   Der Bookmark muss explizit durch Anklicken im Bookmark-Dialog gestartet werden und ist nicht automatisch aktiv. Das ist der Default Wert.

Start Mode Durch den Start Mode wird festgelegt ob die Query sofort nach dem Start des Bookmarks ausgeführt und das Navigationsfenster angezeigt werden soll oder ob der Query Dialog angezeigt werden soll. Es gibt zwei Ausprägungen:

1. NAVIGATE

   Die Query wird sofort ausgeführt und der Navigationsbildschirm mit allen Ergebnissen angezeigt.

2. QUERY

   Der Query Dialog wird zuerst angezeigt.

21.4 Detail Navigation

Hat man auf einen der Master Jobs (welcher Children hat) geklickt, erscheint das ”Detail” Fenster des Master Jobs. Dies sieht aus wie im Bild 21.5.

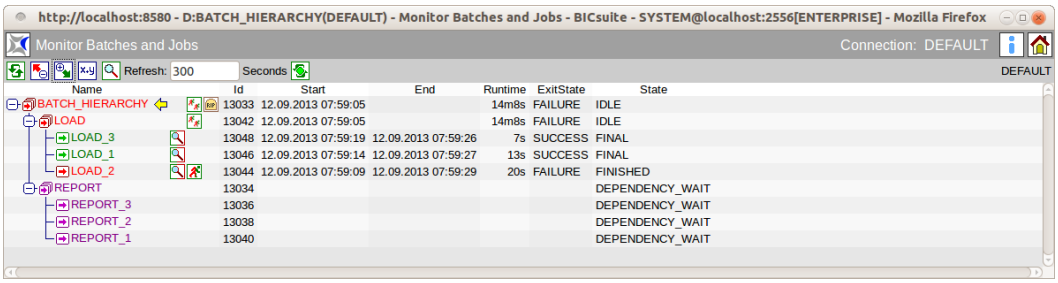


Abbildung 21.5: Running Master Jobs Detailfenster

Das ”Detail” Fenster ähnelt dem Master Fenster. Folgende Unterschiede sind festzustellen:

21.4.1 Tree Darstellung

Statt der flachen Liste im Master Fenster, werden alle Children des aktuellen Batches und deren Children in einer Tree Hierarchy-Darstellung angezeigt. Die erste Ebene ist aufgeklappt, jede weitere Ebene nicht. Ausnahme hiervon stellt die Suche mittels Name Patterns dar, da hier jeder gefundene Job aufgelistet wird und dazu die jeweilige Ebene und die darüberliegenden Ebenen ebenfalls automatisch aufgeklappt werden.

21.4.2 Suchergebnis

Das Ergebnis einer Suche wird mittels des Pfeilsymbols



dargestellt. Hierdurch können gefundene Einträge die aufgeklappt wurden, da ein Child-Eintrag den Suchkriterien entspricht, von Treffern unterschieden werden. Nur Treffer werden mittels des Pfeilsymbols angezeigt.

21.4.3 Detail Navigation Query Maske

Die Query Maske sieht aus wie im Bild 21.6.
Die Query Maske für die Detail Navigation entspricht der Query Maske für Master Navigation, mit folgenden Unterschieden:
Die Felder *History*, *Unit*, *Detail Bookmark* und *Autostart* entfallen und es gibt einige zusätzliche Felder.
Folgende Felder sind auf der Maske zu sehen:

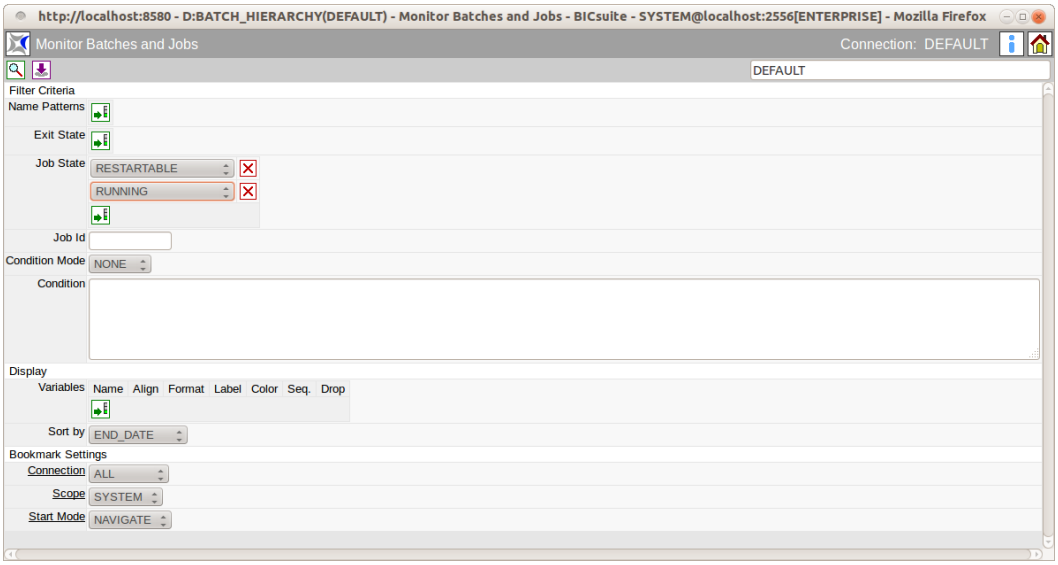


Abbildung 21.6: Detail Navigation Query Maske

Liste Job State In der Liste Job State kann nach Jobs gefiltert werden, welche den eingetragenen Job State haben. Es kann jeder, der im Feld *State* beschriebene, State ausgewählt und nach ihm gefiltert werden.
Werden keine Job States eingetragen, werden diese bei der Suche nicht berücksichtigt.

Job ID In diesem Feld kann eine konkrete Job Id eingetragen werden, nach der gesucht werden soll. Als Ergebnis kann immer nur maximal ein Job gefunden werden.

Sort By Hiermit kann die Liste nach verschiedenen Kriterien sortiert werden. Es sind folgende Kriterien möglich.

1. Name

   Die Ergebnisliste soll alphabetisch nach dem Namen des Jobs sortiert werden.

2. Start Date

   Die Ergebnisliste soll chronologisch nach dem Zeitpunkt des Start des Jobs sortiert werden.

3. End Date

   Die Ergebnisliste soll chronologisch nach dem Zeitpunkt des beenden des Jobs sortiert werden.

21.5 Detailmaske für Jobs

Klickt man im ”Detail” oder im ”Master” Fenster einen Job ohne Children an, wird automatisch die Detailmaske für Jobs angezeigt. Sie sieht folgendermaßen aus:

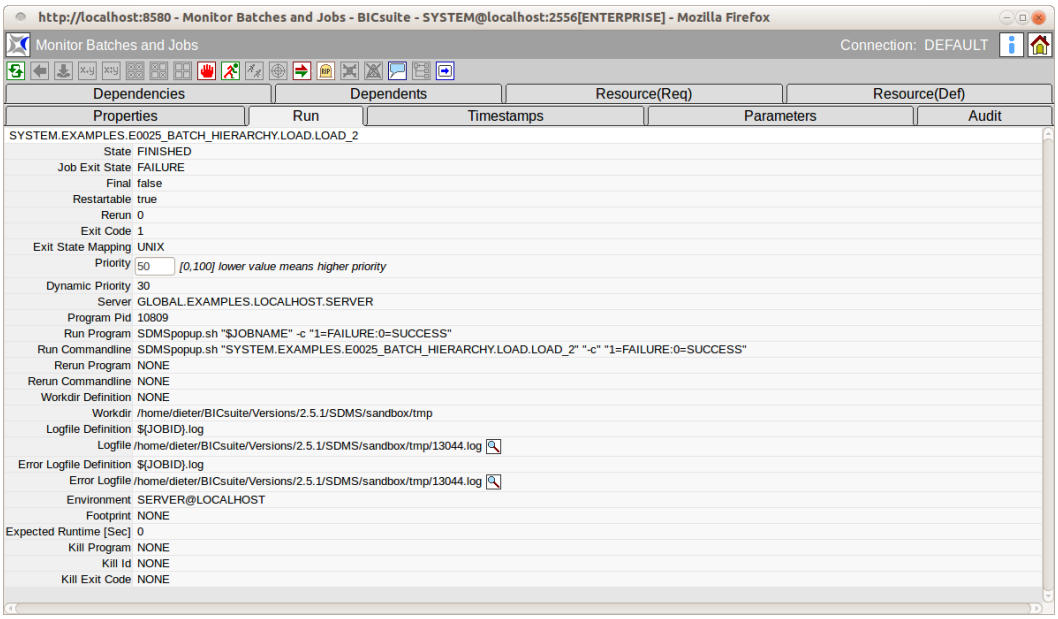


Abbildung 21.7: Detailmaske für Jobs

In dieser Maske werden alle Laufzeitinformationen angezeigt und manche können verändert werden. Des weiteren stehen Aktionsbuttons zur Verfügung, die eine Manipulation an den aktuellen Objekten zulassen.

21.5.1 Buttons

Die folgenden Buttons sind alle Aktionsbuttons. Das heißt, durch diese Buttons wird eine bestimmte Aktion auf dem gewählten Job durchgeführt. Um dies zu erreichen, wird nach der Betätigung des Buttons der folgende Confirm Bildschirm angezeigt:
Im Textfeld kann nun ein Kommentar eingegeben werden, welcher den Grund der Aktion oder zusätzliche Bemerkungen beschreibt. Dieser Text wird später im Tab Audit, im Feld *Reason/Comment* angezeigt. Durch erneutes Betätigen des jeweiligen Aktionsbuttons, wird die Aktion durchgeführt. Durch Betätigen des *Cancel* Buttons, wird die Aktion abgebrochen und zum vorherigen Bildschirm zurückgesprungen. Abhängig von der Aktion enthält der Confirm Bildschirm weitere Eingabefelder zur Steuerung der Aktion.

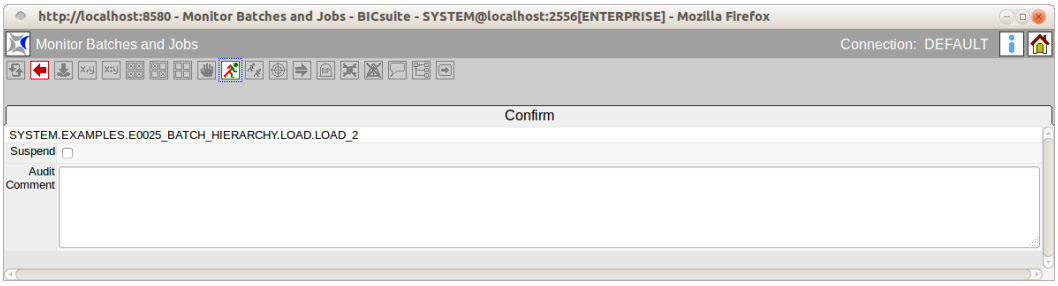


Abbildung 21.8: Confirm Maske

 Suspend

Mittels des *Suspend* Buttons, können Jobs Suspended werden. Dies ist nur möglich falls die Jobs sich nicht schon im State "Suspended" befinden und noch nicht beendet sind. Nach dem erfolgreichen Suspend haben die Jobs den State "Suspended".
Zusätzliche Eingabefelder: *Resume*, *Resume Time*, *Resume In*, *Unit*

Resume Hier kann ausgewählt werden, ob ein automatischer Resume stattfinden soll.
Es gibt folgende Möglichkeiten:

- NO: wählt diese Funktionalität ab und es werden keine anderen Eingabefelder angezeigt.
- AT: wählt einen automatischen Resume zu einem festen Zeitpunkt. Das Eingabefeld *Resume Time* wird angezeigt.
- IN: wählt einen automatischen Resume nach Ablauf einer Zeit. Die Eingabefelder *Resume In* und *Unit* werden angezeigt.

Resume Time Hier wird der gewünschte Resume Zeitpunkt im Format "YYYYMMDD HH:MM" eingegeben.

Resume In Hier wird angegeben, wie viele Zeiteinheiten (siehe *Unit*) bis zum Resume gewartet werden soll.

Unit Hier wird eingegeben ob es sich bei der Eingabe im *Resume In* um Minuten (MINUTE), Stunden (HOUR), oder Tage (DAY) handeln soll.

 Resume

Mittels des *Resume* Buttons können Jobs, welche sich im State "Suspended" befinden, wieder aktiviert werden. Das heißt, es erfolgt eine erneute Prüfung der Ressourcen und Abhängigkeiten und falls alle Prüfungen erfolgreich sind, wird der Job auf einem Jobserver gestartet.
Zusätzliche Eingabefelder: *Resume*, *Resume Time*, *Resume In*, *Unit*

Resume Hier kann die Art des Resumes spezifiziert werden.

- OFF: Steht nur zur Auswahl falls der Job suspended und ein Autoresume (IN/AT) aktiv ist. Diese Option deaktiviert diesen Autoresume.
- NOW: Es findet ein sofortiger Resume statt. Es werden keine anderen Eingabefelder angezeigt.
- AT: wählt einen automatischen Resume zu einem festen Zeitpunkt. Das Eingabefeld *Resume Time* wird angezeigt.
- IN: wählt einen automatischen Resume nach Ablauf einer Zeit. Die Eingabefelder *Resume In* und *Unit* werden angezeigt.

Resume Time, Resume In, Unit Siehe Oben (Aktion Suspend)

 Rerun
Der *Rerun* Button erscheint, falls der aktuelle Job beendet wurde und sich in einen "Restartable" Zustand befindet. Das heißt, er besitzt einen Exit State welcher nicht Final ist. Durch Drücken des *Rerun* Buttons, wird das Rerun Programm aufgerufen (falls definiert, ansonsten das Run-Kommando). Hiermit ist ein erneutes Starten eines fehlerhaften Jobs, nach der Problembehebung möglich.
Zusätzliche Eingabefelder: *Suspended*, *Resume*, *Resume Time*, *Resume In*, *Unit*

Suspended Wird diese Flag gesetzt, so wird der Job suspended und das Auswahlfeld *Resume* wird angezeigt. Ist dieses Feld nicht gesetzt, so werden keine weiteren Felder angezeigt.

Resume, Resume Time, Resume In, Unit Siehe Oben (Aktion Suspend)

 Rerun Children
Der *Rerun Children* Button wird angezeigt, wenn sich Children des aktuellen Jobs in einem "Restartable" Zustand befinden. Das heißt, sie besitzen einen Restartable Exit State. Durch Drücken des *Rerun Children* Buttons, wird das Rerun für alle Children gestartet, welche Restartable sind. Hiermit ist ein erneutes Starten fehlerhafter Children nach einer Problembehebung möglich.
Zusätzliche Eingabefelder: Siehe Rerun

 Kill

Der *Kill* Button erscheint nur, falls der aktuelle Job im Moment läuft (State Running) und in der Job Definition ein Kill-Programm angegeben wurde. Durch Drücken des Buttons *Kill* wird das jeweilige Kill Program aufgerufen und der State des Jobs auf "TO KILL" umgesetzt. Wird das Kill Program erfolgreich durchgeführt, ändert sich der State des Jobs auf "KILLED". Hiermit ist ein Terminieren des Jobs im Programmlauf möglich.

 Set State

Ermöglicht das manuelle Setzen eines Exit States für einen Job. Der *Set State* Button erscheint, falls ein Job mindestens eine der folgenden Bedingungen erfüllt:

- Der Job ist in einem Fehler Status (restartable)
- Der Job ist in einem PENDING Status
- Der Job ist suspended und nicht aktiv und nicht in einem FINAL Status

Zusätzliche Eingabefelder: *Set Exit State*, *Force*, *Resume*

Set Exit State Hier kann im Feld *Set Exit State* ein gültiger Exit State ausgewählt werden.

Force Hier wird ausgewählt, ob man einen Exit State zulassen will, der nicht vom Job selbst über seinen Exit Code erreicht werden kann.

Resume Hier wird ausgewählt, ob ein suspended Job nach dem Set State resumed werden soll.

 Cancel Run

Der *Cancel Run* Button erscheint, falls der Job auf Ressourcen oder Abhängigkeiten wartet oder noch auf einem Jobserver zugeteilt wurde. Durch Drücken des *Cancel Run* Buttons kann der geplante Joblauf abgebrochen werden. Der Job bekommt den State "Cancelled". Es ist nun kein Rerun und kein Resume möglich. Der Job bekommt auch keinen Exit State. Jobs, welche auf einen gecancelten Job warten, müssen diesen Job ignorieren oder ebenfalls gecancelt werden.

 Clear Warning

Der Button *Clear Warning* erscheint, wenn eine Warnung für den Job vorliegt. Mit dem *Clear Warning* kann diese Warnung gelöscht werden, so dass das Warning Icon für den Job in der Übersichtsliste nicht mehr angezeigt wird.

 Comment

Mittels des *Comment* Buttons, kann ein beliebiger Kommentar bzgl. des Jobs eingetragen werden. Es wird außer dem Kommentareintrag keine weitere Aktion am Job durchgeführt.

 Edit Job

Beim *Edit Job* Button handelt es sich nicht um einen Aktionsbutton, da hier keine Änderung am aktuellen Laufzeitobjekt durchgeführt werden kann. Wird der *Edit Job* Button betätigt, erscheint ein neues Fenster mit dem Batches and Jobs Definitionsscreen der Job Definition. Die angezeigten Daten beziehen sich auf die Definition des Ablaufobjektes zum Zeitpunkt des Submits.

21.5.2 Tab Properties

Der Tab Properties enthält alle Daten bzgl. der Job Definition und des aktuellen States. Er sieht folgendermaßen aus:

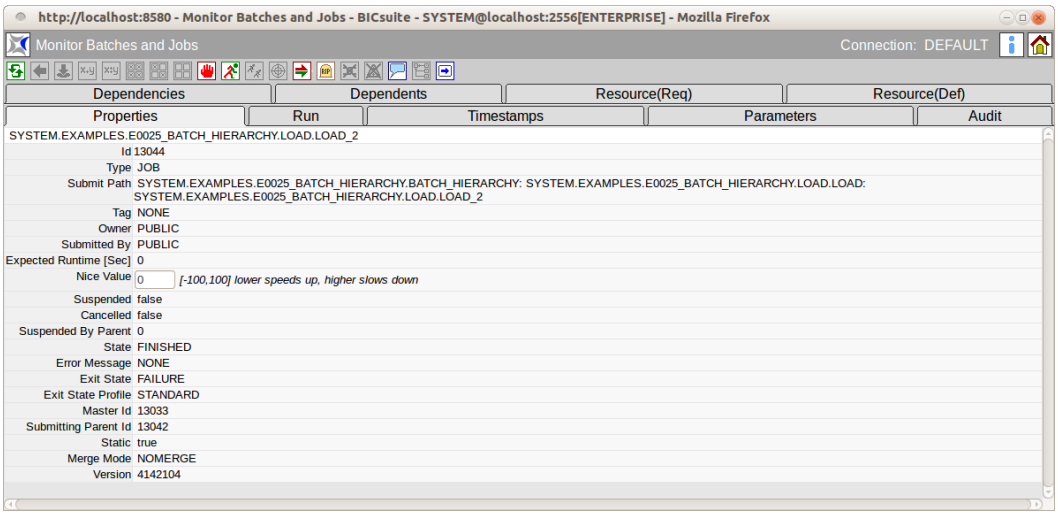


Abbildung 21.9: Job und Batch Properties

Die Felder des Tabs ”Properties” haben folgende Bedeutung:

ID Das ist die eindeutige Identifikationsnummer des Joblaufzeitobjektes.

Type Das ist der Typ des Laufzeitobjektes. Mehr zu Typen finden Sie im Kapitel 14.5.1.

Submit Path Das ist der vollständige Submit-Pfad des Laufzeitobjektes. Das heißt, hier wird kommagetrennt angegeben, über welchen Pfad der Submit stattgefunden hat.

Tag Das ist der Name des Child Tags. Dies ist nur bei einem dynamischen Child notwendig und gibt eine eindeutige Identifikation der aktuellen Child-Instanz. Mehr zu dynamischen Children finden Sie im Kapitel 14.5.4.1.

Owner Das ist der Name der Gruppe, welche die Job Definition für dieses Laufzeitobjekt erstellt hat.

Submitted By Das ist der Name der Gruppe, welche die Laufzeitinstanz submitted hat. Wurde der Job dynamisch durch ein Parent-Programm submitted, ist dies der Benutzer, welcher den Parent submitted hat.

Unresolved Handling Das Feld *Unresolved Handling* ist nur in der Maske des Master Jobs sichtbar. Es zeigt die Einstellung des Feldes *On Unresolved Error* aus der Submit-Maske.

- Ignore: Die Unresolved Dependencies werden grundsätzlich ignoriert.
- Suspend: Im Falle von Fehler, aufgrund unresolved Dependencies, wird der gesamte Ablauf ”suspended” submitted
- None: Default-Verhalten, im Falle von unresolved Dependencies, wird gehandelt wie in der Dependency Definition vorgegeben.

Nice Value Bei diesem Feld handelt es sich um ein Eingabefeld. Der Nice Value kann hier eingegeben werden. Mehr zum Nice Value finden Sie im Kapitel 14.5.4.1. Da es sich hier um das einzige Eingabefeld des Dialoges handelt, muss die Tabulatortaste nach der Eingabe betätigt werden, damit der *Save* Button aktiv wird um die Änderung abzuspeichern.

Suspended Das Feld gibt an ob das Laufzeitobjekt aktuell Suspended ist (TRUE) oder nicht (FALSE). Durch die Buttons *Suspend* und *Resume* kann der Suspend State geändert werden.

Suspend by Parent Gibt an ob das Laufzeitobjekt durch den Parent Suspended wurde (1) oder nicht (0).

State Zeigt den aktuellen State des Laufzeitobjektes an. Mehr zum State finden Sie im Kapitel 21.3.

Error Message Wurde im Feld *State* der State "ERROR" gemeldet, wird eine nähere Beschreibung des Fehlers genannt.

Exit State Ist der Job beendet, wird hier der resultierende Exit State des Jobs mitgeteilt.

Exit State Profile Hier wird das Exit State Profile, welches in der Job Definition eingetragen wurde, angezeigt. Mehr zum Exit State Profile finden Sie im Kapitel 14.5.1.

Master Id Hierbei handelt es sich um die ID des Master Jobs, welcher submitted wurde, um dieses Laufzeitobjekt zu erzeugen. Ist das Objekt selbst als Master Job submitted worden, ist diese identisch mit der ID.

Submitting Parent ID Hierbei handelt es sich um die ID des Parent-Laufzeitobjektes, welcher den aktuellen Job submitted hat. Hat der Job kein Parent, wird "NONE" angezeigt.

Static *Static* ist das Kennzeichen das angibt ob es sich beim aktuellen Laufzeitobjekt um einen statisch (TRUE) oder um ein dynamisch submittetes Child (FALSE) handelt. Mehr zu dynamischen Children finden Sie im Kapitel 14.5.4.1.

Merge Mode Gibt den aktuell verwendeten Merge Mode an. Mehr zum Merge Mode finden Sie im Kapitel 14.5.4.1.

Version Gibt die Versionsnummer, der beim Submit aktuellen Job Definition, an.

21.5.3 Tab Run

Der Tab Run enthält alle Informationen, welche sich auf die aktuellen Informationen zum Run Program und zur Laufzeitumgebung und des Rechners beziehen. Der Tab sieht folgendermaßen aus:
Die Felder des Tabs "Run" haben folgende Bedeutung:

State Zeigt den aktuellen State des Laufzeitobjektes an. Mehr zum State finden Sie im Kapitel 21.3.

Job Exit State Der Job Exit State enthält den aktuellen Exit State des Jobs falls der Job beendet ist, ansonsten "NONE".

Final Handelt es sich bei dem Job Exit State um einen Final State, dann steht der Wert des Feldes auf "TRUE", ist der Job Exit State "NONE" oder kein Final State, dann steht der Wert des Feldes auf "FALSE".

Detailmaske für Jobs

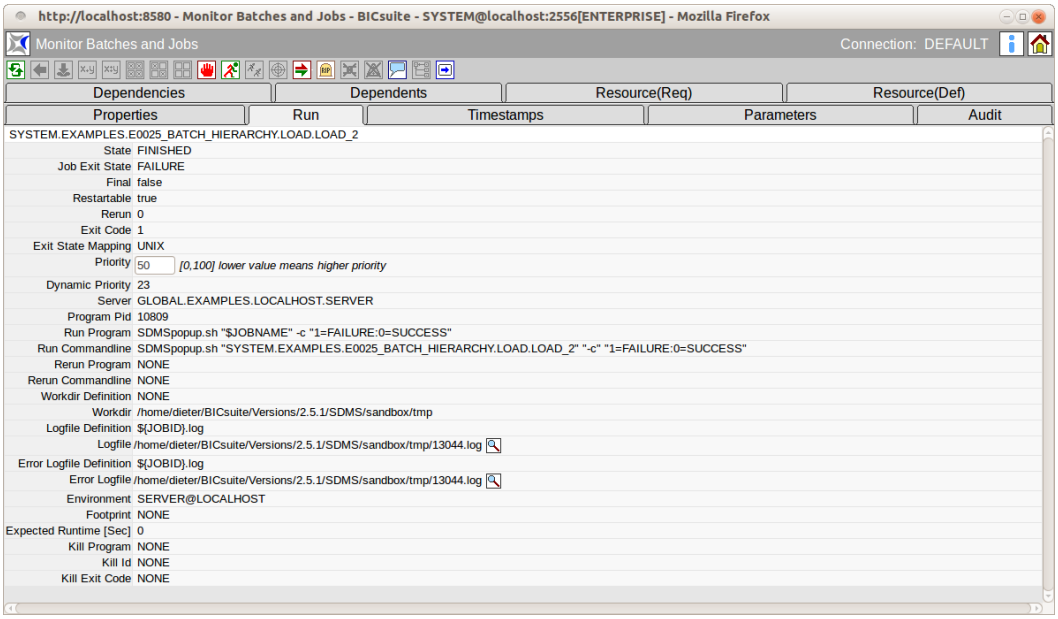


Abbildung 21.10: Batch und Job Run Information

Restartable Ist der Job restartable, dann steht der Wert auf ”TRUE” ansonsten auf ”FALSE”. Falls das Feld den Wert TRUE hat, ist der Button *Rerun* aktiv.

Rerun Im Feld *Rerun* steht die Anzahl der Neustarts des Jobs.

Exit Code Beim Exit Code handelt es sich um den Exit-Wert, den das Run Program bei der Beendigung des Prozesses hatte. Mehr zu Exit Code finden Sie im Kapitel 14.5.2.

Exit State Mapping Das im Batches und Jobs Dialog definierte Exit State Mapping des Jobs, welches beim Umwandeln des Exit Codes in den Job Exit State verwendet wurde. Mehr zu Exit State Mapping finden Sie im Kapitel 14.5.2.

Priority Hierbei handelt es sich um ein Eingabefeld. Es wird die aktuelle Priorität angezeigt und diese kann verändert werden. Durch anschließendes Betätigen der Tabulatortaste und Drücken des *Save* Buttons wird die neue Priorität gespeichert. Mehr zum Thema Priorität finden Sie im Kapitel 14.5.2.

Dynamic Priority Hierbei handelt es sich um die dynamische (effektive) Priorität des Jobs, welche bei ansteigender Zeit vom Submit bis zum Ausführen steigt. Mehr zur dynamischen Priorität finden Sie im Kapitel 13.4.2.3.

Server Das Feld *Server* gibt den aktuellen Jobserver an, auf welchem der Prozess ausgeführt wird oder wurde.

Program PID Die *Program PID* ist die Prozess ID des Run Programs.

Run Program Hierbei handelt es sich um die im Batches-And-Jobs-Dialog definierte Kommandozeile des Run Programs. Variablen und Parameter werden hier noch mit ihren Namen angezeigt und werden nicht ersetzt.

Run Commandline Die im Batches and Jobs Dialog definierte Kommandozeile des Run Programs mit ersetzten Werten. Das ist die Darstellung, wie sie der zu verarbeitenden Shell übergeben wurde. Alle Variablen und Parameter wurden mit ihren aktuellen Werten ersetzt.

Rerun Program Die im Batches-And-Jobs-Dialog definierte Kommandozeile des Rerun Programs. Variablen und Parameter werden hier noch mit ihren Namen angezeigt und werden nicht ersetzt. Wurde kein Rerun Program definiert, steht hier NONE.

Rerun Commandline Die im Batches and Jobs Dialog definierte Kommandozeile des Rerun Programs, mit ersetzten Werten. Das ist die Darstellung, wie sie der zu verarbeitenden Shell übergeben wurde. Alle Variablen und Parameter wurden mit ihren aktuellen Werten ersetzt. Ist kein Rerun durchgeführt worden, steht hier NONE.

Workdir Definition Im Feld *Workdir Definition* ist das Verzeichnis von dem der Prozess ausgeführt wird angezeigt. Steht hier ”None”, wird das default Arbeitsverzeichnis des ausführenden Jobservers genutzt.

Workdir Hierbei handelt es sich um das im aktuellen Job verwendete bzw. zu verwendende Workdir. Ist der Job beendet oder läuft er aktuell, werden die ersetzten Parameter und Variablen angezeigt, ansonsten die Parameter.

Logfile Definition Die Logfile Definition entspricht der Definition des Logfiles im Batches and Jobs Dialog.

Logfile Hier steht der Name und der Pfad des aktuell verwendeten Logfiles (falls der Job beendet ist oder noch läuft), ansonsten NONE. Mögliche Variablen und Parameter werden durch die tatsächlichen Werte ersetzt.
Mittels des Buttons


kann ein neues Fenster aufgemacht werden, welcher den Inhalt des Logfiles darstellt. Sollte dies nicht möglich sein, wenden Sie sich bitte an Ihren Systemadministrator.

Error Logfile Definition Die Error Logfile Definition entspricht der Definition des Error Logfiles im Batches and Jobs Dialog.

Error Logfile Hier steht der Name und der Pfad des aktuell verwendeten Error Logfiles (falls der Job beendet ist oder noch läuft) ansonsten NONE. Mögliche Variablen und Parameter werden durch die tatsächlichen Werte ersetzt.
Der Button

hat die gleiche Funktion, wie soeben bei Logfile beschrieben.

Environment Das Environment entspricht der Definition des Environments im Batches and Jobs Dialog.

Footprint Der Footprint entspricht der Definition des Footprints im Batches and Jobs Dialog.

Expected Runtime Die Expected Runtime entspricht der Definition im Batches and Jobs Dialog.

Kill Program Das Kill Program entspricht der Definition des Kill Programs im Batches and Jobs Dialog.

Kill Id Die Kill ID enthält die Prozess ID des Kill Programs, falls dieses gestartet wurde. Wurde noch kein Kill durchgeführt ist die Kill ID NONE.

Kill Exit Code Der Kill Exit Code ist das Ergebnis, welches das Kill Program nach Beendigung des Kill-Vorganges an den Jobserver zurückliefert. Hiermit kann der Erfolg oder Misserfolg eines Kill-Versuches ermittelt werden, falls dies vom Kill Program vorgesehen ist.

21.5.4 Tab Timestamps

Der Tab Timestamps gibt verschiedene Zeitpunkte für den jeweiligen Job an. Mit diesem Zeitpunkt kann ein Überblick über das Zeitverhalten und die Dauer der Jobausführung gegeben werden. Der Tab sieht folgendermaßen aus:
Der Tab ist nur informativ, es können keine Eingaben gemacht werden.
Die Felder des Tabs ”Timestamps” haben folgende Bedeutung:

Detailmaske für Jobs

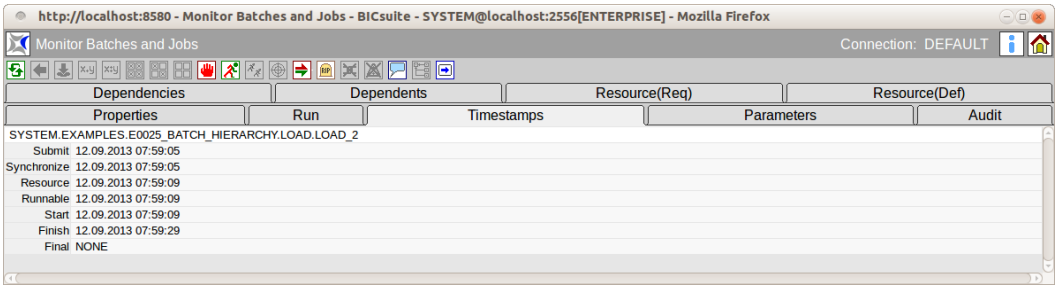


Abbildung 21.11: Batch und Job Timestamps

Submit Im Timestamp Submit steht die Zeit des Submit-Zeitpunktes für den aktuellen Job. Handelt es sich um einen statischen Job, ist dies der Submit-Zeitpunkt für den Master Job, bei dynamischen Jobs, die Zeit an dem der Parent Job den dynamischen Submit durchgeführt hat. Der Submit-Zeitpunkt stimmt mit dem Zeitpunkt, zu dem der Job an den State Dependency Wait gewechselt hat, überein.

Synchronize Der Timestamp Synchronize zeigt an, zu welchem Zeitpunkt der Job in den State "Synchronize Wait" übergegangen ist. Das heißt, wann alle möglichen Abhängigkeiten des Jobs erfüllt wurden.

Resource Im Timestamp Resource wird die Zeit des Statusüberganges in Resource Wait angezeigt. Das heißt, zu diesem Zeitpunkt wurden alle geforderten Synchronizing Resources erfüllt und dem Job zugeteilt.

Runnable Der Zeitpunkt Runnable gibt an, wann der Job in den State "Runnable" eingetreten ist. Das heißt, alle Dependencies und Resource-Anforderungen konnten erfüllt werden. Ab jetzt kann der Job von einem Jobserver abgearbeitet werden.

Start Im Timestamp Start steht die Zeit des Starts des Run Programs durch den Jobserver. Sollte der Job schon mehrmals ausgeführt worden sein (Rerun), ist hier der letzte Start eingetragen.

Finish Im Timestamp Finish steht die Zeit der Beendigung des Jobs. Sollte der Job schon mehrmals ausgeführt worden sein (Rerun), ist hier die letzte Beendigung eingetragen.

Final Im Timestamp Final steht die Zeit der Beendigung des Jobs, die einen Final State erzeugt hat. Wurde der Job beendet, besitzt aber einen Restartable State, wird hier NONE eingetragen.

21.5.5 Tab Dependencies

Im Tab Dependencies werden alle Abhängigkeiten des aktuellen Jobs zu anderen Jobs angezeigt. Der Tab sieht folgendermaßen aus:

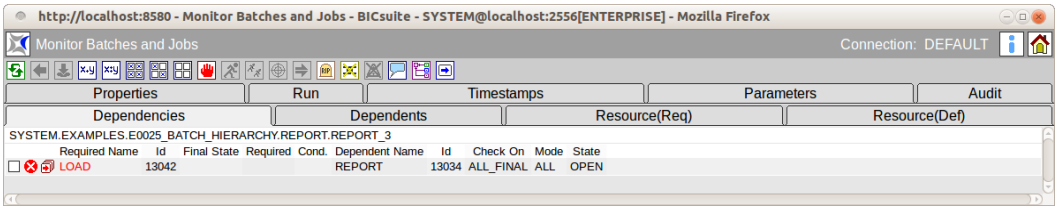


Abbildung 21.12: Batch und Job Abhängigkeiten

Der Dialog gibt eine Übersicht über alle notwendigen Abhängigkeiten, ihren aktuellen State und die Möglichkeit Abhängigkeiten bewusst zu ignorieren.
Folgende Buttons sind auf der Maske sichtbar:

 Ignore Dependencies

Der Button *Ignore Dependencies* wird verwendet um eine oder mehrere Abhängigkeiten bewusst zu ignorieren. Alle zu ignorierenden Abhängigkeiten müssen vorher in der Maske ausgewählt werden. Anschließend wird der Button gedrückt. Nun erscheint ein neues Fenster zur Auswahl des Ignore Modus und zur Eingabe eines Vermerks. Das Fenster sieht folgendermaßen aus:

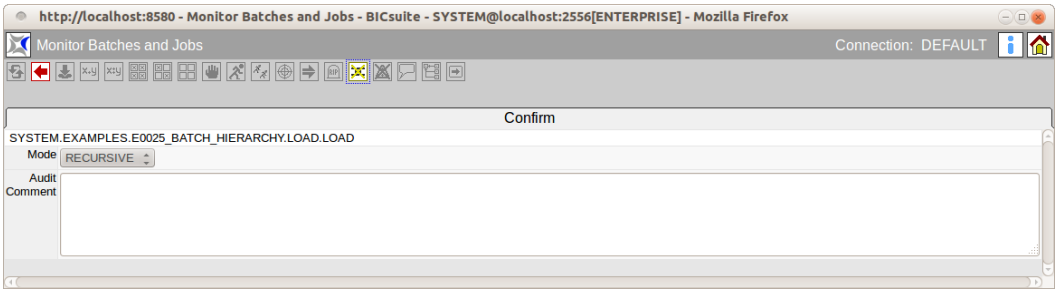


Abbildung 21.13: Batch und Job Ignore Dependencies Confirm Maske

Auf dieser Maske muss der Modus des Ignore eingegeben werden. Per Default erben alle Children eines Submitted Entities dessen Abhängigkeiten: wird B von A benötigt, so ist auch jedes Child von A (CA) von B abhängig. Mittels des Modus des Ignore ist nun das Verhalten der vererbten Abhängigkeit bei den Children bestimmbar.
Es gibt folgende Optionen:

1. Recursive

Dies ist das Default-Verhalten.

Das Ignore bezieht sich nicht nur auf den aktuellen Job, sondern auch auf die vererbten Abhängigkeiten, die Children des aktuellen Jobs zu dem benötigten Job haben. Das heißt, der Parent Job ignoriert diese Abhängigkeit und alle seine Children ignorieren diese vererbte Abhängigkeit ebenfalls. Sollte ein Child eine explizit angegebene Abhängigkeit, welche nicht vererbt wurde, zum benötigten Job haben, so wird diese nicht ignoriert sondern beachtet.

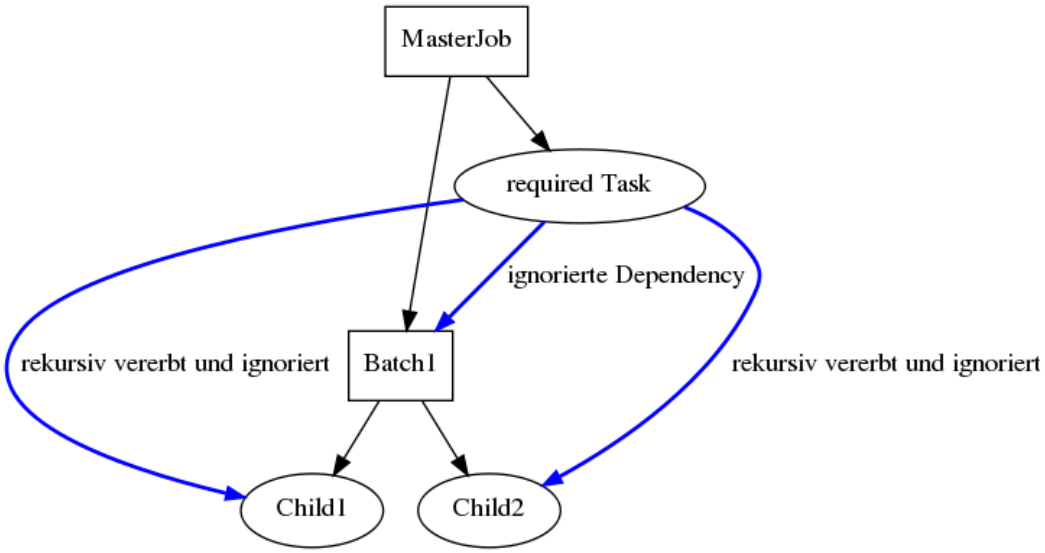


Abbildung 21.14: Beispiel für Recursive Ignore

2. Job Only

   Nur der Job selbst ignoriert die Abhängigkeit. Alle Children beachten die vererbte Abhängigkeit und warten auf den benötigten Job. Damit ist es möglich den Parent Job laufen, seine Children allerdings warten zu lassen.

Nach der Eingabe des Ignore Modus und der Eingabe eines Kommentars im Kommentarfeld, kann durch ein erneutes Betätigen des *Ingore Dependency* Buttons die Aktion durchgeführt werden.
Die Liste des Tabs zeigt alle Dependencies, welche zum Starten des Jobs erfüllt werden müssen, und ihren aktuellen State an. Innerhalb der Liste kann eine Selektion stattfinden, die mittels des Buttons *Ignore Dependencies* zum Ignorieren der Abhängigkeiten benutzt werden kann.
Folgende Felder sind in der Liste sichtbar:

Icon-Feld Über das *Icon* Feld ist der aktuelle State der Abhängigkeit ersichtlich. Es gibt folgende Möglichkeiten:

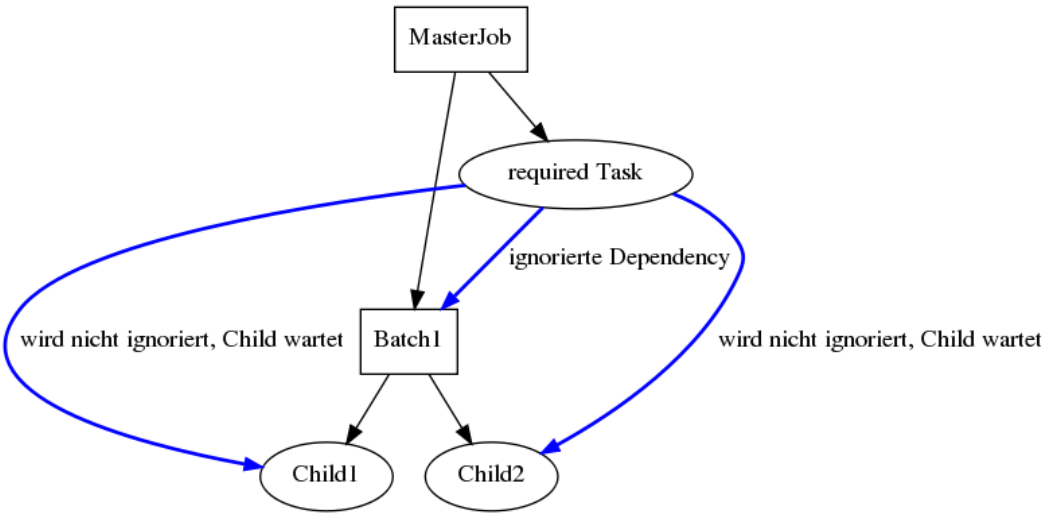


Abbildung 21.15: Beispiel für Job Only Ignore



Die Abhängigkeit wurde noch nicht erfüllt oder kann nicht mehr erfüllt werden. Sollte bei einer Abhängigkeit mit dem Dependency Mode ALL dieses Icon zu sehen sein, kann durch diese Abhängigkeit der aktuelle Job nicht ausgeführt werden. Im Dependency Mode ANY muss mindestens eine Abhängigkeit nicht Rot sein, damit der Job ablaufen kann.



Die Abhängigkeit wurde erfüllt. Im Dependency Mode ANY muss mindestens eine Abhängigkeit den State grün haben, damit der Job laufen kann. Im Dependency Mode ALL müssen alle Einträge grün sein, damit der Job laufen kann.

Required Name Hierbei handelt es sich um den Namen des Jobs, welcher als Voraussetzung für den Start des aktuellen Jobs gelaufen sein muss. Durch Anklicken des Namens, kann zu seiner Laufzeitdefinition gewechselt und dessen State überprüft werden.

ID In diesem Feld wir die Laufzeit-ID des Required Jobs angezeigt.

Final State Wurde der Required Job abgearbeitet, steht hier der finale State des Jobs. Ist hier ein Final State eingetragen und die Abhängigkeit ist nicht erfüllt, muss der Grund ein falscher Final State für diese Abhängigkeit sein. Das heißt, es gibt einen Unterschied zwischen Final State und Required State. Soll der Job nun doch

ausgeführt werden, kann dies nur durch das Ignorieren über den Ignore *Dependency* Button geschehen. Befinden sich die Abhängigkeiten der Liste im Dependency Mode ANY, muss mindestens eine andere Abhängigkeit erfüllt sein.

Required In diesem Feld wird der benötigte Final State des Jobs angezeigt. Damit die Abhängigkeit erfüllt ist, muss der Final State identisch mit dem Required State sein oder der Required State NONE sein.

Dependent Name In diesem Feld wird der Job genannt, welcher diese Abhängigkeit fordert. Dies ist im Normalfall der aktuell gewählte Job, es kann aber auch einer der Parents des Jobs sein.

Check On Das Feld *Check On* beschreibt das Verhältnis der Beendigung des Required Jobs. Es gibt folgende Ausprägungen:

1. All Final

   Der Required Job und alle sein Children müssen Final sein

2. Job Only

   Nur der Required Job muss final sein, die Children werden nicht beachtet.

Mehr zum Thema Check On finden Sie im Kapitel 14.5.6.1.

ID Hierbei handelt es sich um die ID des abhängigen Jobs.

Mode Der *Mode* legt fest, wie die Liste der Jobs zu beachten ist. Es gibt folgende Ausprägungen:

1. All

   Es müssen alle Abhängigkeiten der Liste erfüllt werden, damit der Job starten kann.

2. Any

   Es muss mindestens eine Abhängigkeit der Liste erfüllt werden, damit der Job starten kann.

Mehr zum Mode finden Sie im Kapitel 14.5.6.

State Hierbei handelt es sich um den aktuellen State der Abhängigkeitsbeziehung. Es gibt folgende Ausprägungen:

1. OPEN

   Der Required Job hat noch keinen Final State erreicht (ist noch nicht gelaufen, läuft noch, ist auf Fehler gelaufen). Die Abhängigkeit ist noch nicht erfüllt, kann aber durch Erreichen des richtigen Final States erfüllt werden.

2. FULFILLED

   Der Required Job ist beendet und hat den notwendigen Final State erreicht. Die Abhängigkeit ist erfüllt worden.

3. FAILED

   Der Required Job ist beendet und hat einen falschen Final State erreicht. Die Abhängigkeit kann nicht mehr erfüllt werden. Im Dependency Mode ALL kann der aktuelle Job nur durch Ignorieren der Dependency zum Laufen gebracht werden. Im Dependency Mode ANY muss mindestens eine der anderen Abhängigkeiten erfüllt werden.

21.5.6 Tab Dependents

Im Tab Dependents werden alle Abhängigkeiten von anderen Jobs zum aktuellen Jobs angezeigt. Der Tab sieht folgendermaßen aus:

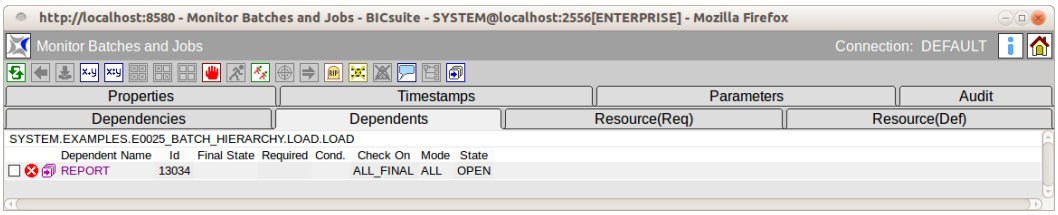


Abbildung 21.16: Batch und Job Dependents Tab

Der Dialog gibt eine Übersicht über alle bestehenden Abhängigkeiten, ihren aktuellen State und die Möglichkeit Abhängigkeiten bewusst zu ignorieren.
Folgende Buttons sind auf der Maske sichtbar:

 Ignore Dependencies

Der Button *Ignore Dependencies* wird verwendet um eine oder mehrere Abhängigkeiten bewusst zu ignorieren. Alle zu ignorierenden Abhängigkeiten müssen vorher in der Maske ausgewählt werden. Anschließend wird der Button gedrückt. Nun erscheint ein neues Fenster zur Auswahl des Ignore Modus und zur Eingabe eines Vermerks. Das Fenster sieht folgendermaßen aus:

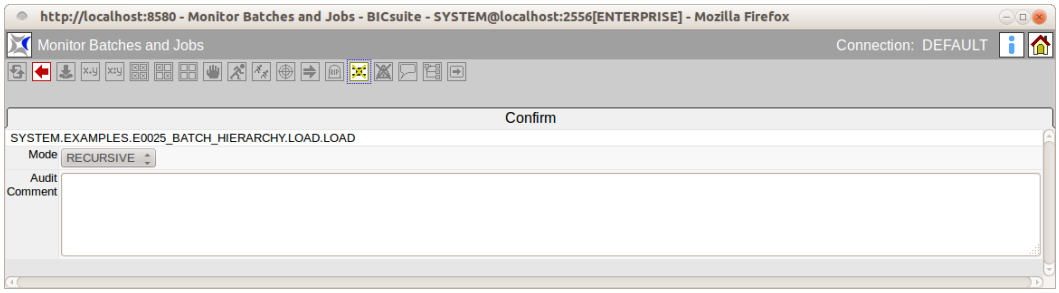


Abbildung 21.17: Batch und Job Ignore Dependencies

Auf dieser Maske muss der Modus des Ignore eingegeben werden. Per Default erben alle Children eines Submitted Entities dessen Abhängigkeiten: wird B von A benötigt, so ist auch jedes Child von A (CA) von B abhängig. Mittels des Modus des Ignore ist nun das Verhalten der vererbten Abhängigkeit bei den Children bestimmbar.
Es gibt folgende Optionen:

1. Recursive

   Dies ist das Default-Verhalten.

   Das Ignore bezieht sich nicht nur auf den abhängigen Job, sondern auch auf die vererbten Abhängigkeiten, die Children des abhängigen Jobs zum aktuellen Job haben. Das heißt, der Parent Job ignoriert diese Abhängigkeit und alle seine Children ignorieren diese vererbte Abhängigkeit ebenfalls. Sollte ein Child eine explizit angegebene Abhängigkeit, welche nicht vererbt wurde, zum benötigten Job haben, so wird diese nicht ignoriert sondern beachtet.

2. Job Only

   Nur der abhängige Job selbst ignoriert die Abhängigkeit. Alle Children beachten die vererbte Abhängigkeit und warten auf den aktuellen Job. Damit ist es möglich den Parent Job laufen, seine Children allerdings warten zu lassen.

Nach der Eingabe des Ignore Modus und der Eingabe eines Kommentars im Kommentarfeld, kann durch ein erneutes Betätigen des *Ingore Dependency* Buttons die Aktion durchgeführt werden.
Die Liste des Tabs zeigt alle Dependencies, welche zum Starten anderer Jobs erfüllt werden müssen, und ihren aktuellen State an. Innerhalb der Liste kann eine Selektion stattfinden, die mittels des Buttons *Ignore Dependencies* zum Ignorieren der Abhängigkeiten benutzt werden kann.
Folgende Felder sind in der Liste sichtbar:

Icon-Feld Über das *Icon* Feld ist der aktuelle State der Abhängigkeit ersichtlich. Es gibt folgende Möglichkeiten:



Die Abhängigkeit wurde noch nicht erfüllt oder kann nicht mehr erfüllt werden. Sollte bei einer Abhängigkeit mit dem Dependency Mode ALL dieses Icon zu sehen sein, kann durch diese Abhängigkeit der abhängige Job nicht ausgeführt werden. Im Dependency Mode ANY muss mindestens irgendeine andere, hier nicht sichtbare Abhängigkeit des abhängigen Jobs nicht Rot sein, damit der abhängige Job laufen kann.



Die Abhängigkeit wurde erfüllt. Im Dependency Mode ANY muss mindestens eine Abhängigkeit den State grün haben, damit der abhängige Job laufen kann. Im Dependency Mode ALL müssen evtl. auch andere, hier nicht sichtbare Abhängigkeit des abhängigen Jobs erfüllt sein, damit der abhängige Job laufen kann.

Dependent Name Hierbei handelt es sich um den Namen des Jobs, welcher davon abhängig ist, dass der aktuelle Jobs gelaufen ist. Durch Anklicken des Namens, kann zu seiner Laufzeitdefinition gewechselt und dessen State überprüft werden.

ID In diesem Feld wir die Laufzeit-ID des abhängigen Jobs angezeigt.

Final State Wurde der abhängige Job bereits abgearbeitet, steht hier der finale State des Jobs.

Required In diesem Feld wird der benötigte Final State des aktuellen Jobs angezeigt.

Cond In diesem Feld wird angezeigt ob eine zusätzliche Bedingung für die Abhängigkeit definiert ist.

Check On Das Feld *Check On* beschreibt das Verhältnis der Beendigung des aktuellen Jobs. Es gibt folgende Ausprägungen:

1. All Final

   Der aktuelle Job und alle sein Children müssen Final sein

2. Job Only

   Nur der aktuelle Job selbst muss final sein, die Children werden nicht beachtet.

Mehr zum Thema Check On finden Sie im Kapitel 14.5.6.1.

Mode Der *Mode* legt fest, wie die Liste der Jobs zu beachten ist. Es gibt folgende Ausprägungen:

1. ALL

   Es müssen alle Abhängigkeiten des abhängigen Jobs erfüllt werden, damit dieser starten kann.

2. ANY

   Es muss mindestens eine Abhängigkeit des abhängigen Jobs erfüllt werden, damit dieser starten kann.

Mehr zum Mode finden Sie im Kapitel 14.5.6.

State Hierbei handelt es sich um den aktuellen State der Abhängigkeitsbeziehung. Es gibt folgende Ausprägungen:

1. OPEN

   Der aktuelle Job hat noch keinen Final State erreicht (ist noch nicht gelaufen, läuft noch, ist auf Fehler gelaufen). Die Abhängigkeit ist noch nicht erfüllt, kann aber durch Erreichen des richtigen Final States erfüllt werden.

2. FULFILLED

   Der aktuelle Job ist beendet und hat den notwendigen Final State erreicht. Die Abhängigkeit ist erfüllt worden.

3. FAILED

   Der aktuelle Job ist beendet und hat einen falschen Final State erreicht. Die Abhängigkeit kann nicht mehr erfüllt werden. Im Dependency Mode ALL kann der abhängige Job nur durch Ignorieren der Dependency zum Laufen gebracht werden. Im Dependency Mode ANY muss mindestens eine andere, hier nicht sichtbare Abhängigkeit des abhängigen Jobs erfüllt werden.

21.5.7 Tab Resource(Req)

Im Tab Resource(Req) werden alle Informationen angezeigt, welche über den aktuellen Zustand der angeforderten Resources Auskunft geben.
Der Tab sieht folgendermaßen aus:
Im Tab ist es möglich alle Resource-Instanzen und deren Zuordnung zu Scopes zu betrachten. Des weiteren besteht die Möglichkeit Anforderungen an Ressourcen zu ignorieren.
Der Button *Ignore Resources* hat folgende Bedeutung:

 Ignore Resources

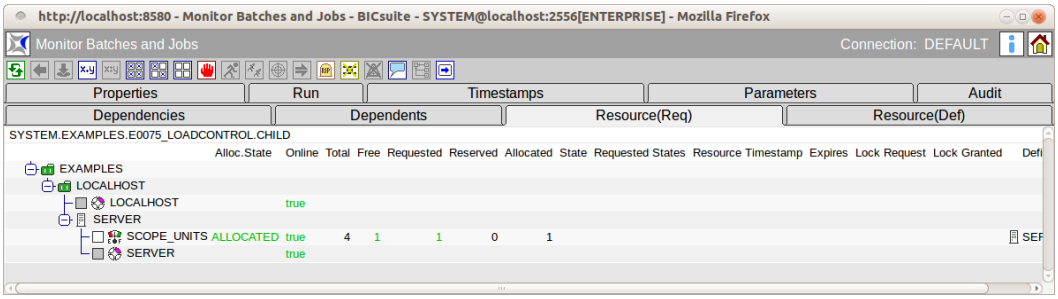


Abbildung 21.18: Job Resource Requirements

Der Button *Ignore Resources* wird verwendet um eine oder mehrere Resource-Anforderungen bewusst zu ignorieren. Alle zu ignorierenden Resource-Anforderungen müssen vorher in der Maske ausgewählt werden. Anschließend wird der Button gedrückt. Nun erscheint eine neue Maske zur Eingabe eines Vermerks. Die Maske sieht folgendermaßen aus:

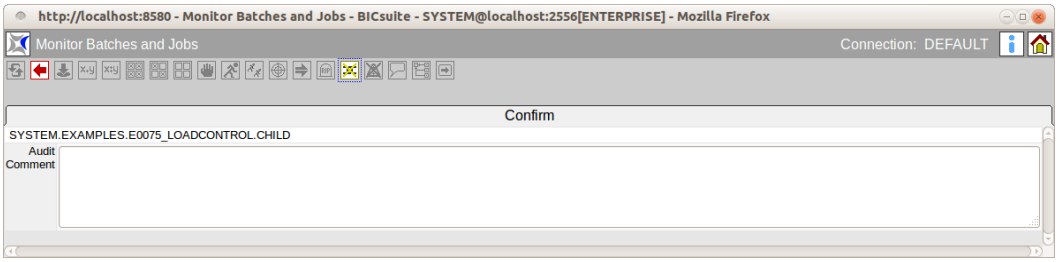


Abbildung 21.19: Job Ignore Resource Requirement

Nach der Eingabe eines Kommentars im Kommentarfeld, wird durch ein erneutes Betätigen des *Ignore Resource* Buttons die Aktion durchgeführt.
In der Liste Resources werden alle angeforderten Ressourcen des Jobs angezeigt. Die Anzeige erfolgt hierarchisch mittels der Scopes und Jobserver, in dem sich die angeforderte Resource befindet. Folgende Spalten sind in der Liste sichtbar:
In der ersten Spalte der Liste werden hierarchisch (sollten sich die Ressourcen innerhalb von Scopes und Jobservern befinden) alle angeforderten Ressourcen mit ihren Namen angezeigt.
Durch einen vorangestellten Auswahlknopf ist es möglich die Ressourcen für ein Ignorieren mittels des Buttons *Ignore Dependencies* zu wählen. Ist das vorangestellte Feld grau, ist dies nicht möglich (bei statischen oder Ressourcen).
Durch das Anklicken des Resource-Namens kann zum Dialog 13.4.2.3 gesprungen werden. Hier ist eine Untersuchung der aktuellen Belegung der Resource möglich.

Allocation State Beim Allocation State handelt es sich um den aktuellen State der Resource-Anforderung. Es gibt folgende Ausprägungen:

1. Keine

   Hierbei handelt es sich um statische oder System Resources, die zum Submit-Zeitpunkt geprüft werden. Sollten diese nicht mehr vorhanden oder offline sein, erfolgt eine Fehlermeldung zur Submit-Zeit.

2. Blocked

   Die Resource wird von einem anderen Job benutzt und kann nicht belegt werden. Der Job wartet nun auf diese Resource. Es kann verschiedene Gründe für eine Blockierung geben. Die Kriterien die eine Belegung verhindern werden in der Liste rot angezeigt.

3. Allocated

   Die Resource wurde vom aktuellen Job belegt und die Anforderung ist damit erfüllt.

4. Available

   Die Resource ist verfügbar, aber noch nicht belegt.

21.5.8 Tab Resource(Def)

Im Tab Resource(Def) werden alle Informationen angezeigt, welche über den aktuellen Zustand der von diesem Submitted Entity definierten Ressourcen Auskunft geben.
Der Tab sieht folgendermaßen aus:

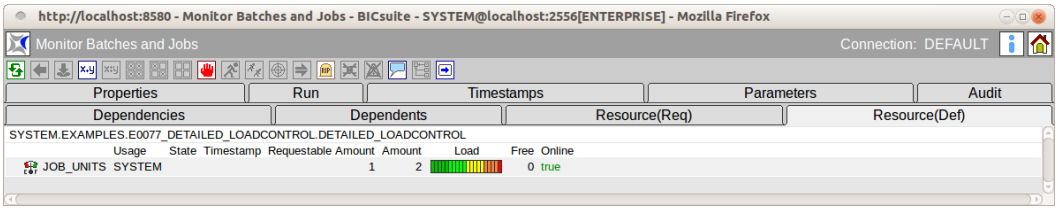


Abbildung 21.20: Batch und Job Defined Resources

In der Liste Resources werden alle definierten Ressourcen des Jobs angezeigt. Folgende Spalten sind in der Liste sichtbar:
In der ersten Spalte der Liste werden alle definierten Ressourcen mit ihren Namen angezeigt.
Durch das Anklicken des Resource-Namens kann zum Dialog 13.4.2.3 gesprungen werden. Hier ist eine Untersuchung der aktuellen Belegung der Resource möglich.

Usage Die *Usage* gibt an, um welchen Typ "Resource" es sich handelt. Mehr zu Typen von Ressourcen finden Sie im Dialog Named Resource.

State Das Feld *State* zeigt den aktuellen Status der Ressource an, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde.

Timestamp er *Timestamp* gibt die Zeit des letzten Statuswechsels einer Resource an, wenn die Usage der Resource "Synchronizing" ist und ein Resource State Profile zugeordnet wurde.

Requestable Amount Wenn die Usage der Resource "Synchronizing" oder "System" ist, gibt das *Requestable Amount* die Menge an, die von einem Job maximal angefordert werden kann.

Amount Wenn die Usage der Resource "Synchronizing" oder "System" ist, gibt das *Amount* die Menge an, die für alles Jobs insgesamt zur Verfügung steht.

Load Wenn die Usage der Resource "Synchronizing" oder "System" ist, zeigt das *Load* die Belegung der Ressource graphisch an.

Free Wenn die Usage der Resource "Synchronizing" oder "System" ist, gibt das *Free* die nicht belegte Menge an, die für alles Jobs noch zur Verfügung steht.

Online Zeigt an ob die Ressource "online" ist, also aktuell angefordert werden kann.

21.5.9 Tab Parameters

Im Tab Parameters werden alle aktuell in diesem Job definierten Ein- und Ausgangsparameter und ihre aktuellen Werte angezeigt. Der Tab sieht folgendermaßen aus:

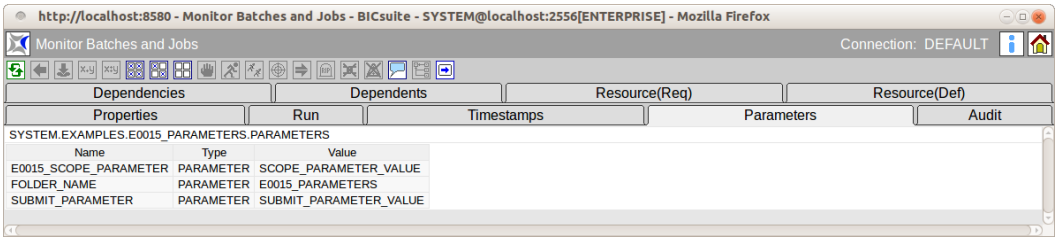


Abbildung 21.21: Batch und Job Parameters

Der Tab dient lediglich zur Information, es sind keine Änderungen möglich.

In der Liste ”Parameter” werden alle Parameter angezeigt, welche aktuell im Job definiert sind.
Informationen zu Parameter finden Sie im Kapitel 14.5.10.

21.5.10 Tab Audit

Im Tab Audit werden alle Informationen über den aktuellen Verlauf der Ausführung des Jobs bzw. Batches und alle durch manuellen Eingriff erzeugten Aktionen und deren Kommentare angezeigt. Dies geschieht in Form einer zeitlich organisierten Liste. Optional können auch alle Audit Informationen von Kindern des Jobs bzw. Batches mit angezeigt werden.
Der Tab sieht folgendermaßen aus:

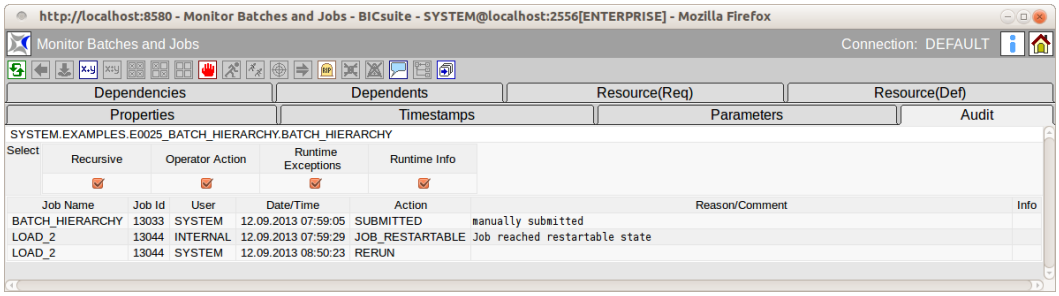


Abbildung 21.22: Batch und Job Auditing

Durch die Checkboxen in der Tabelle ”Select” kann ausgewählt werden, welche Audit Informationen angezeigt werden sollen.

Recursive Ist diese Checkbox gesetzt, werden auch alle Audit-Einträge der Kinder des Jobs bzw. Batches angezeigt.

Operator Action Ist diese Checkbox gesetzt, werden alle Audit-Einträge, welche durch Operator Aktionen erzeugt wurden, angezeigt.
Folgende Aktionen werden angezeigt:

- RERUN RECURSIVE
- CANCEL
- RERUN
- SUSPEND
- RESUME
- SET STATUS

- SET EXIT STATUS
- IGNORE DEPENDENCY
- IGNORE DEPENDENCY RECURSIVE
- IGNORE RESOURCE
- KILL
- COMMENT
- CHANGE PRIORITY
- RENICE
- IGNORE NAMED RESOURCE
- CLEAR WARNING

Runtime Exceptions Ist diese Checkbox gesetzt, werden alle Audit-Einträge, welche durch Ausnahmesituationen im Ablauf erzeugt wurden, angezeigt.
Folgende Aktionen werden angezeigt:

- TRIGGER FAILURE
- RESTARTABLE
- JOB IN ERROR
- SET WARNING
- UNREACHABLE

Runtime Info Ist diese Checkbox gesetzt, werden alle Audit-Einträge, welche durch sonstige Ereignisse im Ablauf erzeugt wurden, angezeigt.
Folgende Aktionen werden angezeigt:

- SUBMIT
- TRIGGER SUBMIT
- SUBMIT SUSPEND
- TIMEOUT
- SET RESOURCE STATUS

Ist ”Operator Action”, ”Runtime Exceptions” und ”Runtime Info” nicht gesetzt, so werden alle Audit-Einträge angezeigt.
Die Tabelle ist rein informativ. Es können keine Änderungen durchgeführt werden. Es wird eine Liste von allen Audit-Einträgen für diesen Job in chronologischer Reihenfolge angezeigt. Folgende Spalten sind auf der Liste sichtbar:

Job Name Der Job Name, zu dem der Audit-Eintrag gehört. Durch einen Klick auf den Job Namen wird der Job bzw. Batch in einem neuen Monitoring Fenster angezeigt.

Job Id Der Job Id, zu dem der Audit-Eintrag gehört. Durch einen Klick auf den Job Namen wird der Job bzw. Batch in einem neuen Monitoring Fenster angezeigt.

User Der Benutzer, der die Aktion für den Audit-Eintrag ausgelöst hat.

Date/Time Der Zeitpunkt, wann die Aktion für den Audit-Eintrag ausgelöst wurde.

Action Die auslösende Aktion. Dies kann eine manuell ausgelöste Aktion sein, aber auch eine automatische Fehlerbehebung etc.

Reason Comment Hier werden die Systemmeldungen bzw. Kommentare die bei manuelle Eingriffe eingegeben wurden angezeigt.

Info Zusätzliche Informationen zur Aktion.

22 Search Running Jobs

22.1 Bild

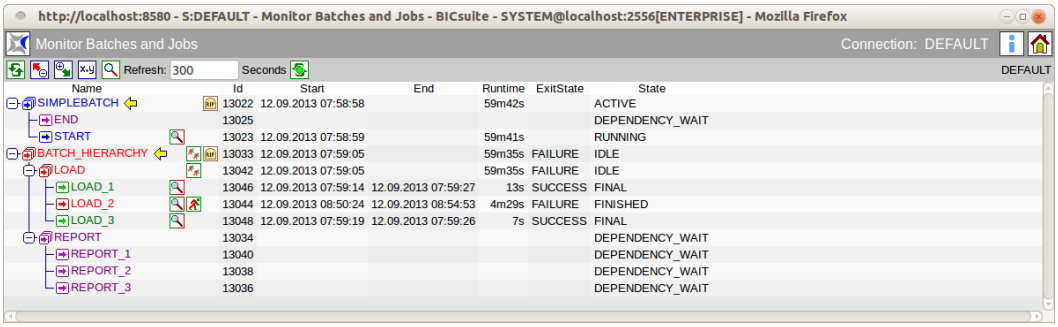


Abbildung 22.1: Search Running Jobs

22.2 Konzept

22.2.1 Kurzbeschreibung

Im Dialog ”Search Running Jobs”, können alle aktuell laufenden oder beendeten Jobs, Batches und Milestones gesucht und angezeigt werden.

22.2.2 Ausführliche Beschreibung

In dieser Konsole ist es möglich alle Jobs zu überprüfen, fehlerhafte neu zu starten, Resources zu ignorieren etc.
Auch die Anzeige Search Running Jobs ist nur eine ”Default” Ansicht (die des Bookmarks ”Default-Search-System”) und kann dementsprechend ebenfalls vom Benutzer angepasst und verändert werden. Wir beschreiben hier die ”Default” Implementierung.

22.3 Navigator

Der Navigationsbildschirm ist analog zur ”Detail” Maske des Running Master Jobs. Weitere Informationen finden Sie dort.

22.4 Detail Navigator Query Maske

Nach dem Aufruf des Dialoges erscheint im Gegensatz zum Running-Master Jobs Dialog zuerst die Query Maske.
Die Query Maske sieht folgendermaßen aus:

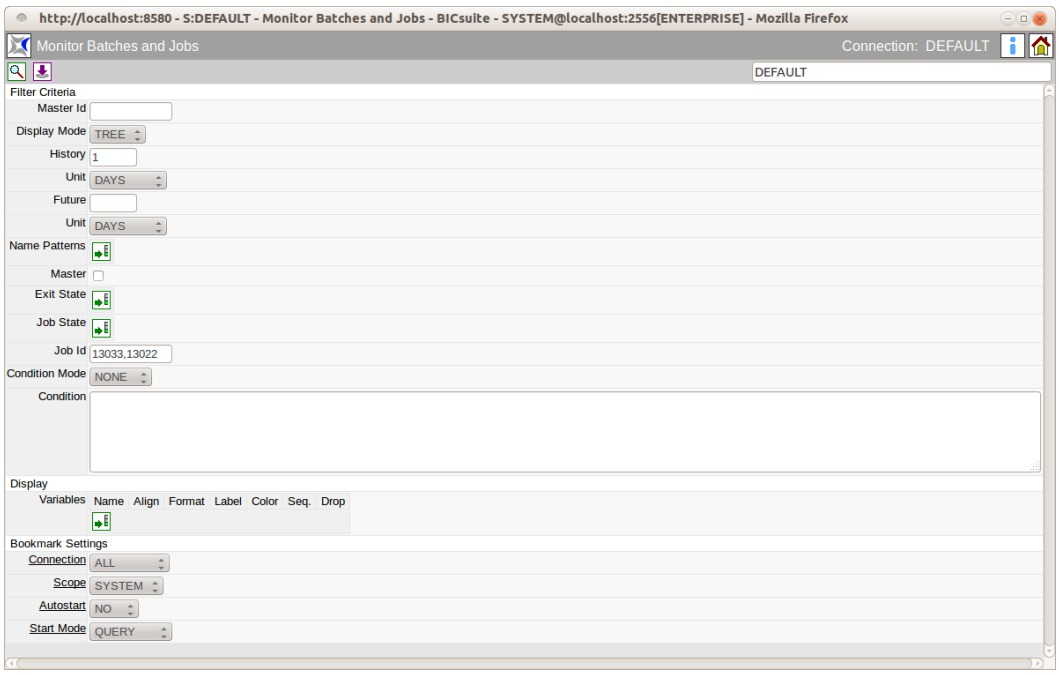


Abbildung 22.2: Search Running Jobs Query Maske

Die Query Maske für die Detail Navigation entspricht dem Query Maske für Running Master Jobs.
Zusätzlich sind folgende Felder definiert:

Master Id Im Feld Master Id kann eine Laufzeit-Id eines Master Jobs eingetragen werden. Alle gefundenen Jobs müssen von dieser Master Id submitted worden sein.

Display Mode Im Feld Display Mode kann die Listendarstellung eingestellt werden. Es gibt folgende Möglichkeiten:

1. LIST

   Die Ausgabe wird als Liste angezeigt.

2. TREE

Die Ausgabe erfolgt hierarchisch nach der Parent-Child-Hierarchie. Es ist möglich in einem Tree zu expandieren und ihn wieder zu schließen.

Master Beim Feld Master handelt es sich um einen Filter, wenn der Schalter gesetzt wurde, mit dem nur Master Submittable Jobs angezeigt werden sollen.

Liste Job State In der Liste Job State kann nach Jobs gefiltert werden, welche den eingetragenen Job State besitzen. Werden keine Job States eingetragen, so werden diese bei der Suche nicht berücksichtigt.

Job ID In diesem Feld kann eine konkrete Job Id eingetragen werden, nach der gesucht werden soll. Als Ergebnis kann immer nur maximal ein Job gefunden werden.

Detail Navigator Query Maske

23 Kalender

23.1 Bild

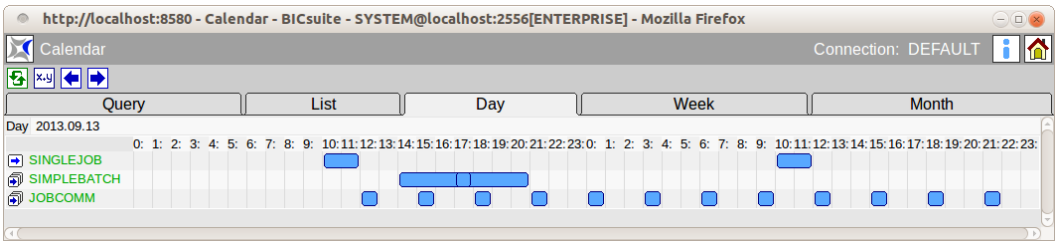


Abbildung 23.1: Kalender

23.2 Konzept

23.2.1 Kurzbeschreibung

Die Kalenderfunktion bietet eine Übersicht über die mittels des Time Scheduling Moduls geplante Ausführungen von Jobs und Batches.
Damit ein Job oder Batch in dieser Übersicht aufgenommen wird, muss beim Anlegen eines Schedules die Kalenderfunktion aktiviert werden (Calendar = active).

23.3 Ausführliche Beschreibung

Der Kalender wird auf unterschiedliche Weise dargestellt. Zum einen gibt es die einfache Liste, chronologisch sortiert, auf der die anstehende Submits aufgeführt werden. Um die Liste nicht unnötig lang, und damit unübersichtlich, zu machen, kann die Auswertungsperiode geändert werden. Standardmäßig werden erst mal die nächste 24 Stunden angezeigt.
Die zweite Darstellung ist die Tagesdarstellung. Dabei wird der Tag als Periode von 24 Stunden dargestellt. Balken geben an, wann welcher Job gestartet werden soll, und, so weit diese Information vom Benutzer gepflegt wurde, wie lange er dauert.
Die dritte und vierte Darstellung entsprechen der Tagesdarstellung, zeigen die Periode aber mir niedriger Granularität. Die Wochendarstellung zeigt sieben Tagen,

aufgeteilt in jeweils 6-Stunden Intervallen. Die Monatsdarstellung zeigt (maximal) 31 Tage.

In der Wochen- und Monatsdarstellung werden die aus zu führenden Jobs bzw. Batches ebenfalls mit Balken dargestellt. Allerdings ist die Laufzeit der Objekte weniger deutlich sichtbar als in der Tagesdarstellung, da die Balken mindestens so lang sind wie die kleinste Einheit in der jeweiligen Ansicht.

So würde ein Batch, der eine erwartete Laufzeit von acht Stunden hat, in der Tagesansicht auch tatsächlich mit 8 Stunden angezeigt. In der Wochenansicht werden zwei Felder, also 12 Stunden angezeigt. In der Monatsansicht wird ein ganzer Tag belegt.

23.3.1 Tab Query

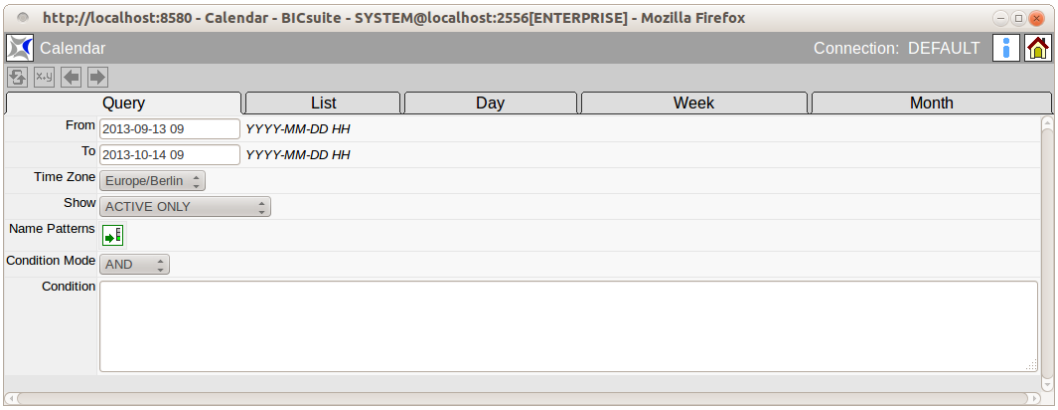


Abbildung 23.2: Kalender Query Tab

Auf dem Tab Query können Suchkriterien für die anzuzeigenden Job Definitions spezifiziert werden. Alle weiter Tabs sind von diesen Einschränkungen betroffen.

Die Felder *From* und *To* Zeile definieren die Periode, für die Startzeiten der anzuzeigenden Jobs und Batches.

Das Feld *Time Zone* erlaubt die Auswahl der Zeitzone in der der Kalender dargestellt werden soll.

In das Feld *Condition* können einfache Bedingungen verknüpft mit Booleschen Operatoren eingegeben werden. In den einfachen Bedingungen können Parameter auf bestimmten Inhalte getestet werden. Die Parameter werden mit **JOB.***parametername* adressiert. Folgendes Beispiel zeigt eine (syntaktisch) gültige Bedingung:

```
job.developer == 'ronald' or
job.developer == 'dieter'
```

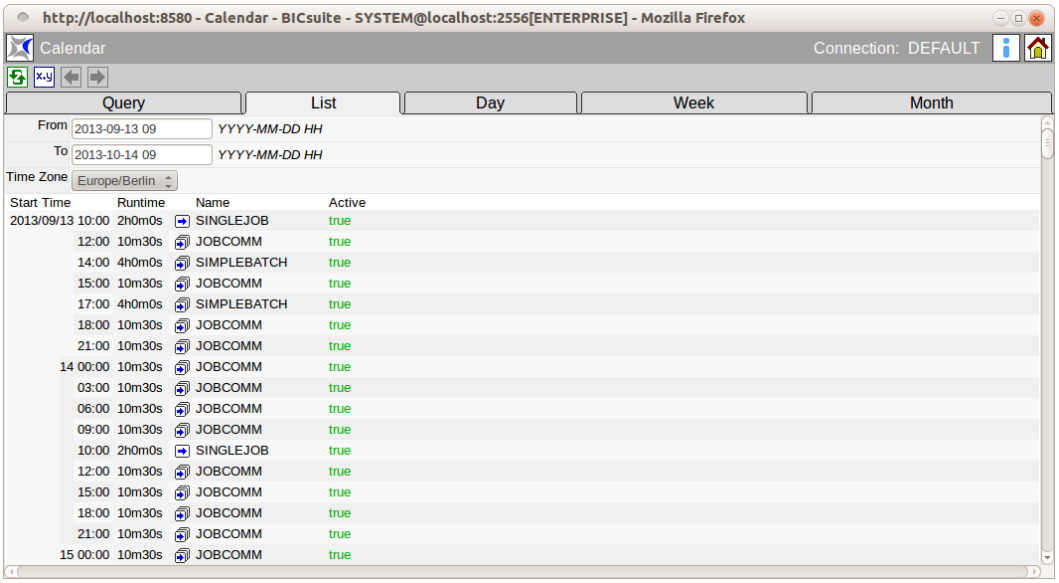


Abbildung 23.3: Kalender Tab List

23.3.2 Tab List

Auf dem Tab "List" werden, für die in den Feldern *From* und *To* eingestellte Periode, alle Startzeiten der eingeplante Jobs und Batches in einer einfachen Liste angezeigt.
Das Feld *Time Zone* erlaubt die Auswahl der Zeitzone in der der Kalender dargestellt werden soll.
In der Spalte *Starttime* steht die geplante Startzeit. Um die Übersichtlichkeit zu erhöhen, werden in dieser Spalte und Zeile nur die Daten angezeigt, die sich im Vergleich mit der vorherigen Zeile geändert haben. Wenn sich zwei aufeinanderfolgenden Zeilen auf den gleichen Tag und Stunde beziehen, wird in der zweiten Zeile nur der Minutenteil des Zeitpunktes angezeigt.

23.3.3 Tab Day

Auf dem Tab "Day" werden alle geplante Jobs und Batches für den selektierten Tag sowie den Tag darauf graphisch angezeigt.
Falls ein Job oder Batch am Anfang voraussichtlich noch aktiv ist, wird dies wie im Bild in der Zeile für "KEEP_RESOURCE" dargestellt. Auch die Darstellung überlappender Läufe ist in der Zeile sichtbar.

23.3.4 Tab Week

Auf dem Tab "Week" werden alle geplante Jobs und Batches für die selektierte Woche sowie die Woche darauf graphisch angezeigt.

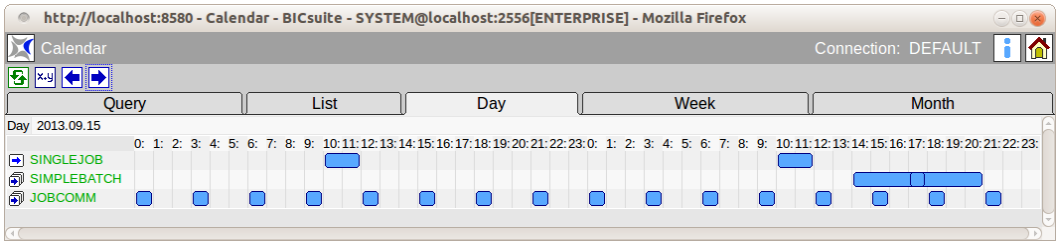


Abbildung 23.4: Kalender Tab Day

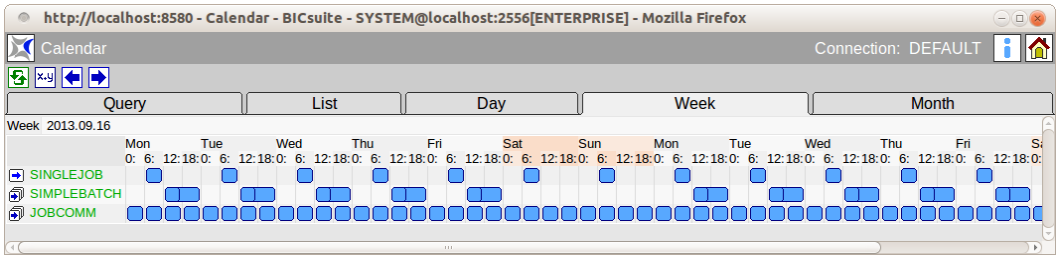


Abbildung 23.5: Kalender Tab Week

23.3.5 Tab Month

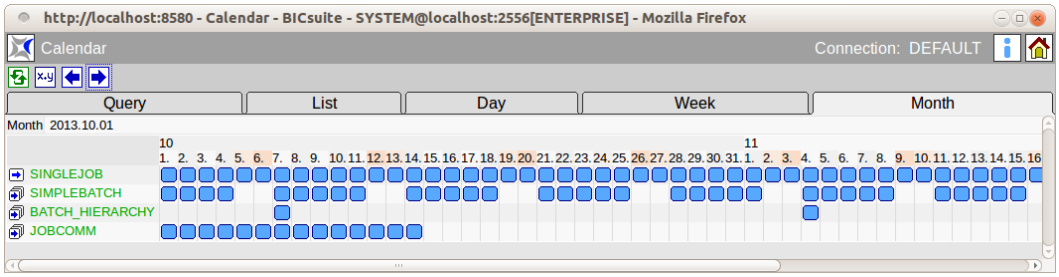


Abbildung 23.6: Kalender Tab Month

Auf dem Tab ”Month” werden alle geplante Jobs und Batches für den selektierten Monat sowie den Monat darauf graphisch angezeigt.

24 System Information

24.1 Bild

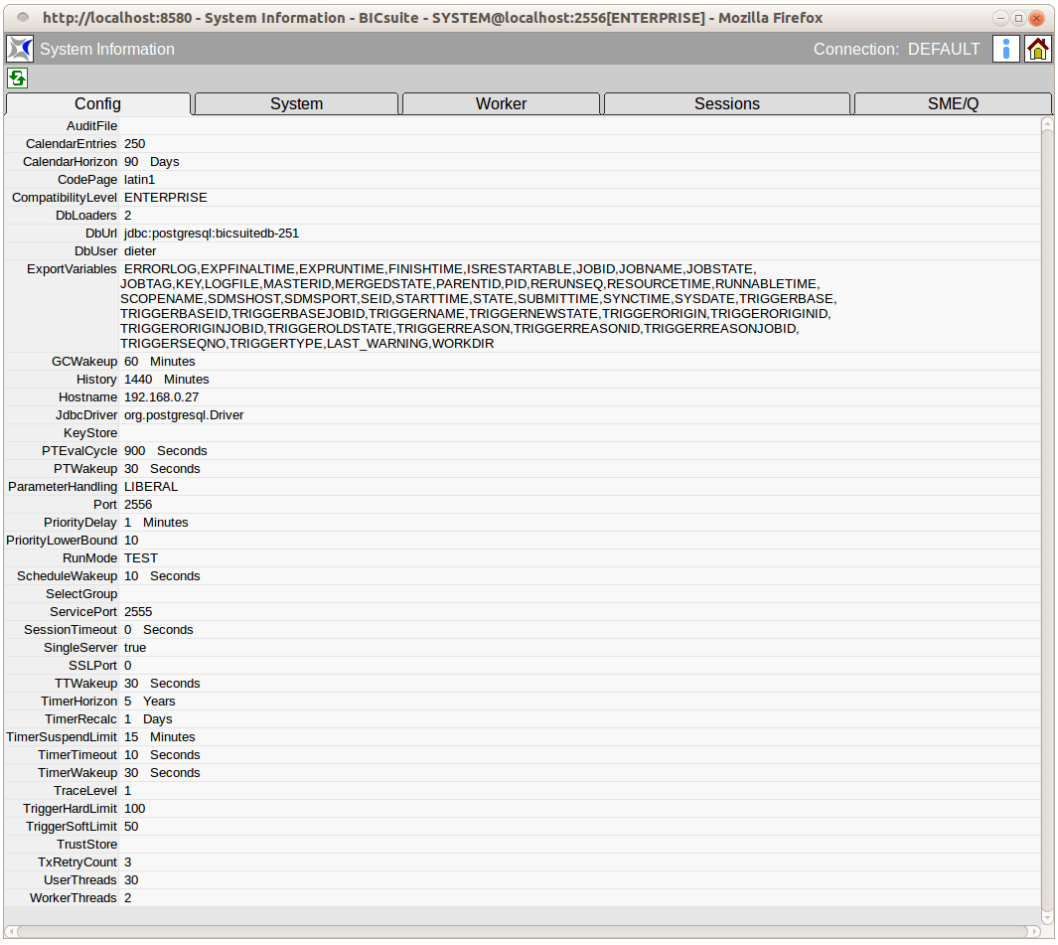


Abbildung 24.1: System Information

24.2 Konzept

24.2.1 Kurzbeschreibung

Der Menüpunkt ”System Information” im Hauptmenü liefert eine Übersicht der Server- und Systemparameter, eine Liste der Workeraktivitäten, sowie eine Liste der aktuellen Sessions.

24.2.2 Tab Config

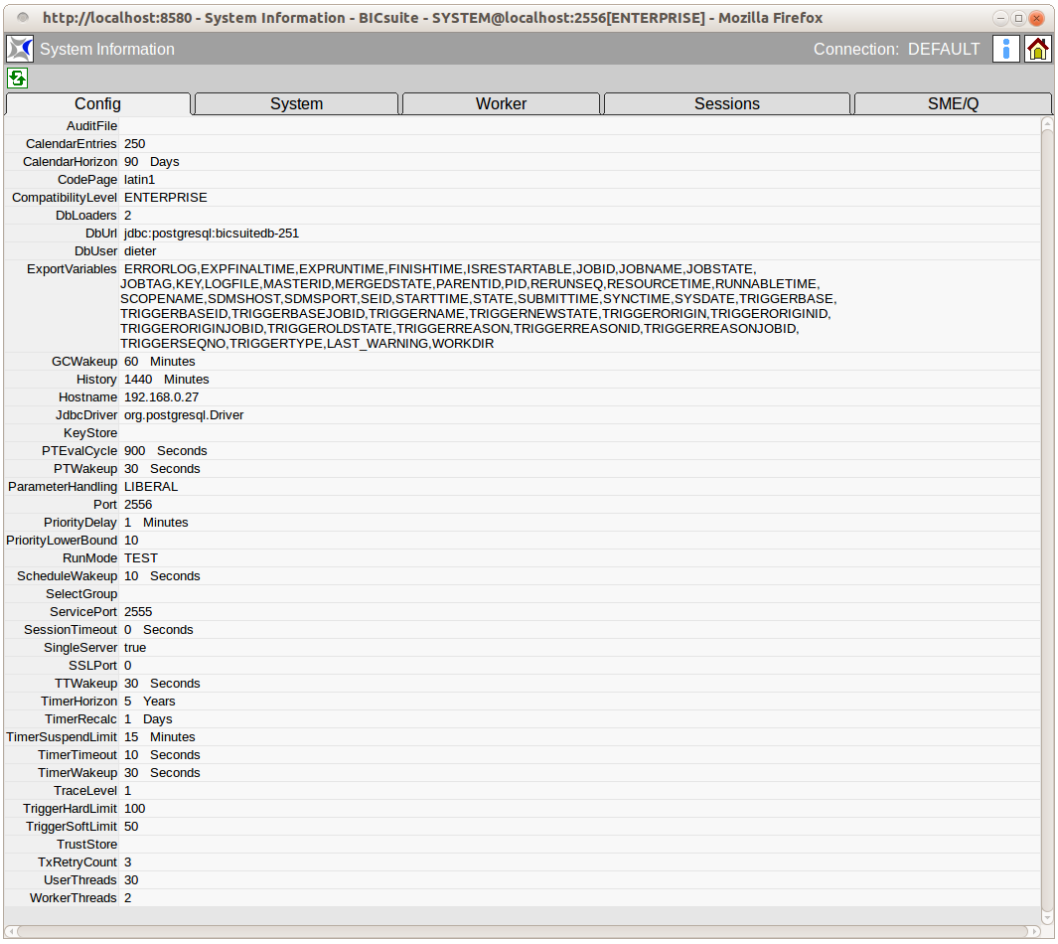


Abbildung 24.2: System Konfiguration

Auf dem Tab Config wird die aktuelle Serverkonfiguration gezeigt. Dabei sind die Parameter in alphabetische Reihenfolge sortiert. Um Parameter zu ändern, muss die Datei `$BICSUITECONFIG/server.conf` angepasst und der Scheduling Server neu gestartet werden. Falls die Umgebungsvariable `BICSUITECONFIG` nicht

gesetzt ist, wird als Konfigurationsverzeichnis `$BICSUITEHOME/etc` verwendet.
Aus Sicherheitsgründen werden nicht alle Serverparameter dargestellt. Insbesondere Zugangsinformation für das Datenbanksystem wird nicht dargestellt.
Die derzeitige Serverparameter sind:

CalendarEntries

Der Parameter **CalendarEntries** definiert die maximale Anzahl Kalendereinträge pro Scheduled Event. Normalerweise wird der Parameter **CalendarHorizon** zuerst Wirkung zeigen, aber für den Fall, dass innerhalb des Horizonts sehr viele Einträge anfallen, verhindert dieser Parameter, dass unsinnig viele Kalendereinträge vorgenommen werden. Da der Horizont pro Schedule definiert werden kann, könnte damit eine Art Denial of Service Angriff vorgenommen werden.
Falls der Parameter nicht definiert wird, gilt als Default Wert 300 Einträge.

CalendarHorizon

Der Parameter **CalendarHorizon** definiert den default Horizont für Kalender.
Falls der Parameter nicht definiert wird, gilt als Default Wert 62 Tage (immer mindestens zwei Monate).

CodePage

Der Parameter **CodePage** definiert den aktiven Zeichensatz. Der Parameter ist inzwischen obsolet.

CompatibilityLevel

Der Parameter **CompatibilityLevel** definiert welche Optionen und Befehle vom Server als gültig erkannt werden. Mögliche Werte sind: **BASIC**, **PROFESSIONAL** und **ENTERPRISE**. Das maximale Level ist abhängig von der lizenzierten Version.
Per Default wird das CompatibilityLevel auf die lizenzierte Version gesetzt.

DbLoaders

Der Parameter **DbLoaders** bestimmt wie viele Threads beim Startup des Servers für das Laden des Objektspeichers gestartet werden. Abhängig von der Anzahl vorhandenen Prozessoren kann eine höhere Anzahl Threads zu einer Beschleunigung des Serverstarts führen.
Als Default wird die Anzahl vorhandenen Prozessoren, jedoch maximal 5 genommen.

DbUrl

Der Parameter **DbUrl** definiert welche Datenbank der Server nutzen soll. Die gültige Syntax ist abhängig vom eingesetzten Datenbanksystem.
Verständlicherweise gibt es für diesen Parameter keine Defaultbelegung.

DbUser

Der Parameter **DbUser** gibt an unter welcher Benutzername mit der Datenbank gearbeitet werden soll.
Auch für diesen Parameter gibt es kein Default.

ExportVariables

Der dargestellte Parameter **ExportVariables** ist eine Zusammensetzung zweier Konfigurationsparameter: **ExportVariables** und **UserExportVariables**. Die genannte Variablen werden beim Abholen eines Jobs als eine Liste von Key-Value Pairs dem Jobserver Übergeben. Diese wiederum hat die Möglichkeit die Variablen in die Umgebung des zu startenden Prozesses zu setzen. Ob der Jobserver dies tut, bzw. unter welchem Name er die Variable in die Umgebung setzt, ist abhängig von der Konfiguration des Jobservers,
Diese Funktionalität wurde auf zwei Parameter aufgeteilt, da die Default Belegung des Parameters **ExportVariables** die Liste der Standardvariablen ist. Um zu verhindern, dass der Administrator die gesamte Liste abschreiben muss, damit nur einen weiteren Parameter exportiert werden kann, gibt es die Möglichkeit, die zusätzliche Variablen im Parameter **UserExportVariables** zu spezifizieren.

GCWakeup

In regelmäßigen Abständen werden "veraltete" Objekte aus dem Objekt-Cache entfernt. Der Parameter **GCWakeup** definiert wie viel Zeit zwischen den Aufräumarbeiten liegen soll. Es ist nicht sinnvoll den Parameter mit einem niedrigen Wert zu belegen, da der Aufwand nur abhängig ist von der Anzahl vorhandenen Objekten (im Wesentlichen die Submitted Entities) und nicht von der Anzahl auf zu räumenden Objekten.
Der Default für diesen Parameter ist 240 (Minuten).

History

Der Parameter **History** definiert die Zeit, für die Information über Master Instanzen von Job bzw. Batch Definitionen behalten wird. Ältere Jobs bzw. Batches werden nur dann im Objekt-Cache behalten, wenn sie noch in irgendeine Weise aktiv, also weder final noch cancelled, sind.
Per Default wird die Information der letzten 10 Tage, also 14400 Minuten behalten.

History Limit

Der Parameter **History Limit** definiert die Zeit, für die Information über nicht mehr aktive Master Instanzen von Job bzw. Batch Definitionen maximal behalten werden. Dieses Limit gilt wenn der Parameter MinHistoryCount ungleich 0 gesetzt ist. Wenn History auf 10 Tage, History Limit auf 30 Tage und MinHistoryCount auf 10 gesetzt ist, werden fr einen wöchentlichen Job bzw. Batch nur die letzten 4 bis 5 Ausführungen im Objekt-Cache gehalten.
Per Default wird die Information der letzten 10 Tage, also 14400 Minuten behalten.

Hostname

Der Parameter **Hostname** ist der Name des Scheduling Servers. Dies ist auch der Inhalt des Standard Job-Parameters **SDMSHOST**.
Per Default wird hier "localhost" gesetzt.

JdbcDriver

Der Parameter **JdbcDriver** definiert welche Klasse als JDBC Driver verwendet wird. Der Name dieser Klasse ist abhängig vom eingesetzten Datenbanksystem. Diese Klasse muss im CLASSPATH auffindbar sein. Die CLASSPATH-Umgebungsvariable wird in die Datei `$BICSUITECONFIG/java.conf` gesetzt.
Es spricht für sich, dass für diesen Parameter kein Default Wert vorgesehen ist.

MinHistoryCount

Der Parameter **MinHistoryCount** definiert die minimale Anzahl von Master Instanzen einer Job bzw. Batch Definition, die im Objekt-Cache gehalten werden. Ein Job bzw. Batch, der nur selten zur Ausführung kommt, kann so deutlich länger als durch den Paremeter **History** definiert im Objekt-Cache gehalten werden. Die maximale Zeit die ein solcher Job bzw. Batch im Objekt-Cache gehalten wird, wird jedoch trotzdem durch den Parameter **History Limit** begrenzt.
Per Default ist der Parameter mit dem Wert 0 belegt und somit nicht aktiv.

MaxHistoryCount

Der Parameter **MaxHistoryCount** definiert die maximale Anzahl von Master Instanzen einer Job bzw. Batch Definition, die im Objekt-Cache gehalten werden. Ein Job bzw. Batch, der sehr häufig zur Ausführung kommt, kann so deutlich kürzer als durch den Paremeter **History** definiert im Objekt-Cache gehalten werden. Ein exzessiver Speicherverbrauch durch solche Jobs bzw. Batches kann dadurch vermieden werden.
Per Default ist der Parameter mit dem Wert 0 belegt und somit nicht aktiv.

PTEvalCycle

Der Parameter **PTEvalCycle** definiert nach wie viele Sekunden die erneute Ressourcenverteilung für einen Pool defaultmäßig erfolgt. Es ist nicht sinnvoll, den Wert für **PTEvalCycle** kleiner als den Wert für **PTWakeup** zu wählen, da die erneute Verteilung erst nach dem Aufwachen des Pool-Threads erfolgt.
Der Default für diesen Parameter ist 900 (Sekunden).

PTWakeup

Der Parameter **PTWakeup** definiert nach wie viele Sekunden der Pool-Thread aufwachen soll um für alle Pools deren **PTEvalCycle** abgelaufen ist eine Neuverteilung der Ressourcen zu errechnen.
Per Default erfolgt alle 300 Sekunden eine Neuverteilung der Ressourcen.

ParameterHandling

Der Parameter **ParameterHandling** definiert wie der Server reagiert, wenn Parameter angesprochen werden, die nicht deklariert wurden. Die mögliche Werte sind:

Wert	**B**edeutung
STRICT	Falls ein nicht deklarierter Parameter benutzt wird, wird eine Fehlermeldung ausgegeben
WARN	Es wird eine Warnung ins Logfile des Servers geschrieben. Der Wert der Variable wird (so weit gefunden) zurückgegeben
LIBERAL	Der Wert des Parameters wird, so weit gefunden, zurückgegeben

Tabelle 24.1: ParameterHandling Optionen

Aus Sicht der Software Engineering sollte **ParameterHandling** als STRICT definiert werden. Aus Sicht der Bequemlichkeit ist WARN oder LIBERAL vor zu ziehen.
Die Default Einstellung ist LIBERAL.

Port

Der Parameter **Port** definiert welcher TCP-Port der Server zur Kommunikation mit den Clients benutzt. Die Wahl des Ports ist frei, so weit kein Port unter 1024 genutzt wird, da nur hoch privilegierte Benutzer dies können. Da der Scheduling Server so konzipiert ist, dass keine besondere Privilegien für den Betrieb benötigt werden, sollte der Server daher auch keine besondere Privilegien erhalten.
Der default Port ist 2506.

PriorityDelay

Der Parameter **PriorityDelay** definiert nach wie viel Zeit die (effektive) Priorität eines Jobs hoch gestuft wird. Die automatische Erhöhung der Priorität verhindert,

dass Jobs unendlich auf Ressourcen warten.
Der Default ist 30 (Minuten).

PriorityLowerBound

Der Parameter **PriorityLowerBound** bestimmt die höchste Priorität die durch normales Altern von Jobs erreicht werden kann. Ist der Parameter größer als 0, dann bleiben eine Anzahl Prioritätsstufen frei, die vom Administrator genutzt werden können, um bestimmte Jobs garantiert vorne in eine Warteschlange ein zu reihen.
Per Default ist der PriorityLowerBound auf 10 gesetzt.

RunMode

Der Parameter **RunMode** sollte in Produktiven Umgebungen auf PRODUCTION gesetzt sein. In der Entwicklungsumgebung von independIT steht der Parameter gelegentlich auf TEST. Falls der RunMode auf TEST steht, ist ein Befehl "RUN TEST ..." freigeschaltet. Es ist nicht definiert welche Auswirkungen die Benutzung dieses Befehls hat, da er für Testzwecke genutzt wird. Es ist sehr wohl möglich, dass die Benutzung des Befehls zu Server-Abstürze führt.
Der RunMode steht auf PRODUCTION so weit er nicht explizit auf TEST gesetzt wurde.

ScheduleWakeup

Der Parameter **ScheduleWakeup** definiert nach wie viel Zeit der Resource Scheduling Thread aufwacht. Da der Resource-Scheduler nahezu keine Zeit beansprucht, wenn es nichts zu tun gibt, kann die Zeit zwischen zwei Aktivierungen relativ kurz gehalten werden.
Der Default Wert beträgt 30 (Sekunden).

SelectGroup

Der Parameter **SelectGroup** definiert welche Gruppe das Select-Statement benutzen darf. Ist dieser Parameter nicht gesetzt, oder enthält er einen Namen einer nicht existierenden Gruppe, dürfen nur Mitglieder der Gruppe ADMIN das Select Statement benutzen.

ServicePort

Der Parameter **ServicePort** ist der Port über den (nur) ein Administrator sich mit dem Server verbinden kann. Dieser Port kann genutzt werden um in Störungsfällen doch noch einen, in der Hauptsache lesenden, Zugriff auf dem Server zu haben.
Der Default Wert ist die 2505.

SessionTimeout

Der Parameter **SessionTimeout** definiert wie lange eine Session per Default idle sein darf, bevor der Server die Session terminiert. Im Normalfall spielt dieser Parameter nur für interaktive Sessions mit `sdmsh` eine Rolle.
Der Default ist 30 (Sekunden).

SingleServer

Der Parameter **SingleServer** ist ein Boolean Wert. Steht er auf "true", geht der Server davon aus, dass kein weiterer Server Zugriff aufs Repository haben kann. Beim Hochfahren nach einem "ungraceful shutdown" ignoriert er gesetzte Tickets und setzt sein eigenes Ticket. Sollte ein zweiter Server unerwarteterweise doch aktiv sein, wird dieser feststellen, dass das Repository jetzt einem anderen Server gehört, und die Arbeit einstellen. Wenn beide Server den Parameter auf 'true" haben, wird eine Art Ping-Pong Effekt die Folge sein.
Die Default Einstellung ist "false".

TTWakeup

Der Parameter **TTWakeup** definiert wie häufig die Tickets im Repository mindestens erneuert werden. Werden mehrere Server eingesetzt, sollten die Parameter auf einander abgestimmt sein (d.h. möglichst gleich).
Der Default ist 30 (Sekunden).

TimerHorizon

Der Parameter **TimerHorizon** definiert wie weit der Timer-Thread in die Zukunft schauen soll um den nächsten Ausführungszeitpunkt eines Jobs zu finden.
Der Default ist 5 (Jahre).

TimerRecalc

Der Parameter **TimerRecalc** bestimmt nach wie viel Zeit erneut versucht wird einen nächsten Ausführungszeitpunkt eines Jobs zu finden, nachdem die letzte Suche aufgrund des Erreichens des TimerHorizons abgebrochen wurde.
Der Default ist 5 (Tage).

TimerSuspendLimit

Der Parameter **TimerSuspendLimit** definiert die Zeit, nach der ein Job suspended submitted wird, falls der Server zur geplanten Submitzeit nicht aktiv war. Diese Grenze kann für jedes Scheduled Event individuell gesetzt werden. Wird dies jedoch unterlassen, gilt der Wert des TimerSuspendLimits.
Die Default Zeit beträgt 15 (Minuten).

TimerWakeup

Der Parameter **TimerWakeup** bestimmt in welchen Abstände der Timer-Thread aktiviert wird. Ein hoher Wert führt zu Ungenauigkeiten bezüglich des tatsächlichen Submitzeitpunktes sowie potentiell lange Laufzeiten des Time-Schedule Vorganges. Ein niedriger Wert ist dagegen deutlich weniger schädlich.
Gute Erfahrungen wurden bis jetzt mit dem default Wert von 30 (Sekunden) gemacht.

TraceLevel

Der Parameter **TraceLevel** bestimmt wie viel und welche Art Logging Information der Server ins Logfile schreibt. Die untenstehende Tabelle zeigt welche Art von Meldungen bei welchem Trace Level protokolliert werden:

Art	**T**race Level	**B**emerkung
FATAL	Alle	FATAL Meldungen werden bei einem schweren Fehlerzustand im Server protokolliert und sollten unverzüglich beim Support-Team von independIT gemeldet werden.
ERROR	Alle	ERROR Meldungen werden bei auftretenden Fehlern protokolliert. Diese Art von Fehlern sind deutlich weniger kritisch als Fehlern der Klasse FATAL. Es ist empfehlenswert diese Art von Fehlern zu untersuchen und wenn möglich ihre Ursachen zu beseitigen.
INFO	Alle	INFO Meldungen geben eine wichtige oder interessante Information aus. Beispiele für solche Meldungen sind die Meldungen die während der Startup-Phase des Servers geschrieben werden.
WARNING	> 0	WARNING Meldungen sind interessante Meldungen die häufig mit Bedienungsungenauigkeiten zu tun haben. Ein Beispiel dafür sind die Warnungen die auftreten wenn der Parameter ParameterHandling auf WARN oder STRICT gesetzt ist.
MESSAGE	> 1	MESSAGE Meldungen sind relativ unwichtige Meldungen, die nichtsdestotrotz sehr interessant sein können. So werden zum Beispiel alle ausgeführte Befehle und ihre Ausführungszeit als MESSAGE protokolliert.

Wird fortgesetzt

Fortsetzung der vorherigen Seite

Art	**T**race Level	**B**emerkung
DEBUG	3	DEBUG Meldungen sind Meldungen die beim Troubleshooting hilfreich sein können. Allerdings führt Trace Level 3 zu einer gewaltigen Flut an Meldungen. Für den Normalbetrieb sind diese Meldungen wenig aussagekräftig und eher behindernd.

Tabelle 24.2: Übersicht der Trace Level

TriggerHardLimit

Der Parameter **TriggerHardLimit** legt die maximale Anzahl, die ein und derselbe Trigger feuern kann, fest.
Die default Einstellung ist 100.

TriggerSoftLimit

Wenn Trigger angelegt werden, ohne dass ein FireLimit spezifiziert wird, legt der Parameter **TriggerSoftLimit** die maximale Anzahl, die eine Instanz dieses Triggers feuern kann, fest.
Die default Einstellung ist 50.

TxRetryCount

In manchen Fällen kann es sein, dass die Durchführung einer Transaktion fehl schlägt, obwohl die Transaktion semantisch und syntaktisch korrekt ist. In dem Fall wird der Server erneut versuchen die Transaktion durch zu führen. Wie oft dies passiert wird definiert durch den Parameter **TxRetryCount**.
Defaultmäßig werden maximal drei Versuche unternommen.

UserThreads

Der Parameter **UserThreads** bestimmt die maximale Anzahl Sessions die gleichzeitig connected sein können. Benutzer die das Web-Frontend benutzen, müssen nicht in der vollen Anzahl mitgezählt werden, da der Zope-Server für jeden Kommunikationsschritt eine neue Verbindung mit dem Scheduling-Server aufbaut und wieder abbaut.
Die Jobserver dagegen bleiben im Normalfall mit dem Scheduling-Server verbunden, und belegen jeweils einen Thread.
Dieser Parameter hat als Defaultwert 10.

WorkerThreads

Der Parameter **WorkerThreads** definiert wie viele Threads für die Befriedigung lesender Abfragen gestartet werden sollen. Im Gegensatz zu den schreibenden Transaktionen sind viele lesende Transaktionen sehr aufwendig. Dabei werden insbesondere die lesende Transaktionen aus dem Monitoring-Bereich genannt.

Per Default werden zwei solche Threads gestartet. In größeren Umgebungen ist es sinnvoll die Zahl zu erhöhen.

24.2.3 Tab System

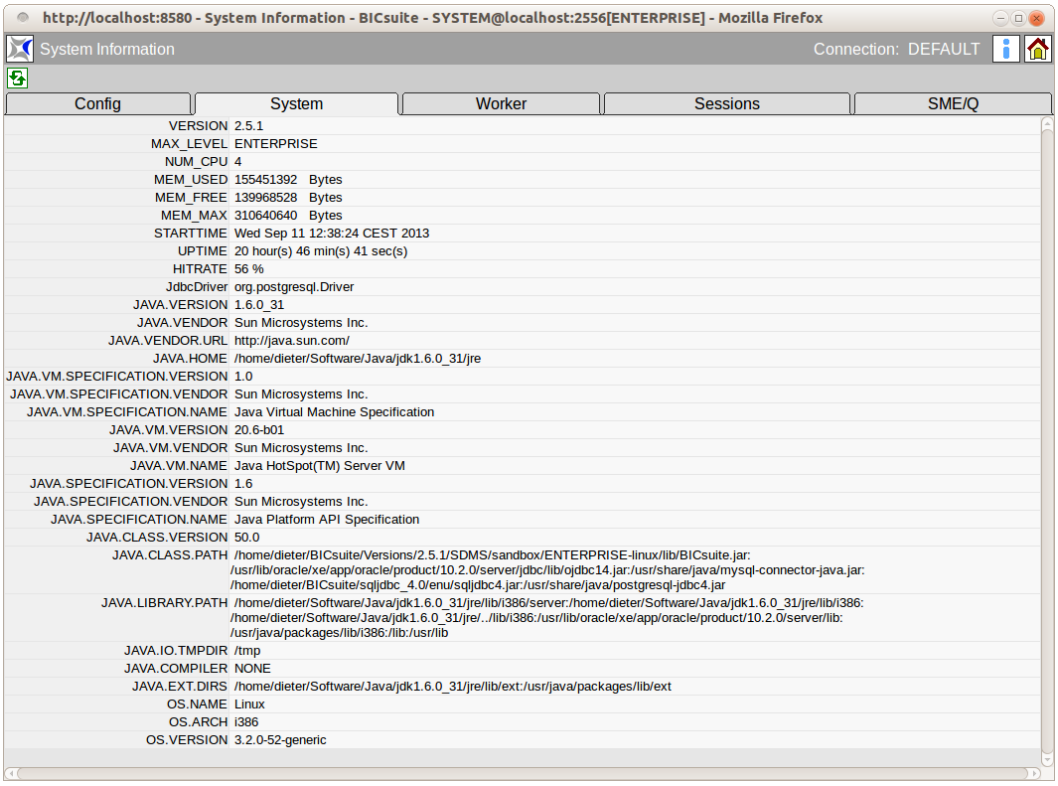


Abbildung 24.3: System Runtime Umgebung

Auf diesem Tab werden einige Laufzeitinformationen sowie Informationen über die Java Virtual Machine dargestellt. Abhängig von der eingesetzten Hardware und dem Betriebssystem bzw. der Java Virtual Machine können letztgenannte Informationen von den dargestellten Informationen abweichen. Es wird deshalb für die korrekte Interpretation dieser Informationen auf die jeweilige Dokumentation verwiesen.

VERSION

Das Feld VERSION zeigt die eingesetzte Software-Version des Scheduling Servers.

MAX_LEVEL

Das Feld MAX_LEVEL zeigt die lizenzierte Version des Scheduling Servers.

NUM_CPU

Das Feld NUM_CPU zeigt die Anzahl von der Java Virtual Machine gemeldeten Prozessoren.

MEM_USED

Das Feld MEM_USED zeigt wie viel Speicher momentan von der Virtual Machine belegt wird.

MEM_FREE

Das Feld MEM_FREE zeigt wie viel des belegten Speichers noch für den Server zur Verfügung steht.

MEM_MAX

Das Feld MEM_MAX zeigt wie viel Speicher die Virtual Machine maximal belegen darf. Falls MEM_MAX und MEM_FREE den gleichen Wert haben, bedeutet das, dass außer der freien Platz gemeldet unter MEM_FREE kein weiterer Speicher mehr zur Verfügung steht. An sich ist das kein Problem. Wenn jedoch die Menge MEM_FREE zuneige geht, wird die Virtual Machine sich immer häufiger um Garbage Collection kümmern müssen, was schwer auf die Performance des Systems lasten kann. In dem Fall ist ein Restart des Servers (mit einer eventuellen Anpassung der Speicherkonfiguration in
`$BICSUITECONFIG/java.conf`) das kleinere Übel.

STARTTIME

Das Feld STARTTIME zeigt den Zeitpunkt des Starts des Servers.

UPTIME

Das Feld UPTIME dient nur der Bequemlichkeit und zeigt wie lange der Server bereits läuft.

HITRATE

Das Feld HITRATE zeigt die Effektivität eines Caching-Algorithmus in Resource Scheduling Thread. Da dieser Cache nur im Fall eines kompletten Reschedule seine Wirkung zeigt, sind bereits relativ niedrige Werte eine Indikation für Effektivität.

JdbcDriver

Der Eintrag hinter dem Titel JdbcDriver wurde auf diesem Tab nur zur Vollständigkeit aufgenommen, da auch die sonstige Java betreffende Konfigurationsparameter hier vorhanden sind. Es handelt sich hier um den selben Wert wie auf dem Tab "Config".

Weiter Einträge

Alle weitere Einträge sind abhängig von der eingesetzten Hardware, vom Betriebssystem und der Java Virtual Machine. Obwohl die Werte in Prinzip relativ leicht interpretierbar sind, wird für eine genaue Beschreibung der Bedeutung auf die Java Dokumentation verwiesen.

24.2.4 Tab Worker

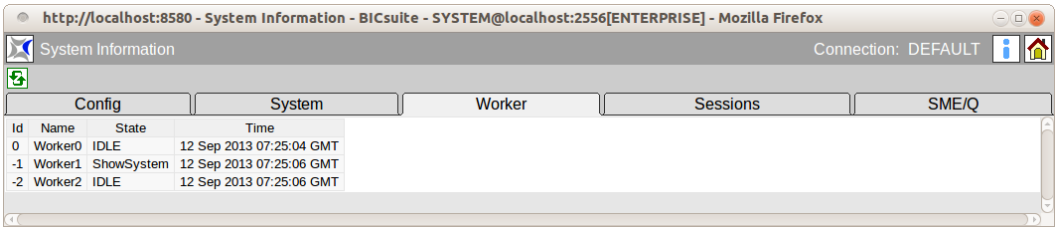


Abbildung 24.4: Worker Thread Information

Auf dem Tab "Worker" wird in einer Tabelle die Aktivität der Worker Threads gezeigt. Der Worker mit der Id 0 ist immer der Worker der für die Verarbeitung der schreibenden Transaktionen zuständig ist. Alle weitere Worker sind zuständig für die Verarbeitung der lesenden Transaktionen.
Sind regelmäßig alle Worker Threads tätig, ist das eine Indikation für eine zu geringe Menge Worker Threads und sollte die Anzahl erhöht werden.

Id

In der Spalte Id steht die interne Nummer des Workers.

Name

In der Spalte Name steht der Name des Workers.

State

In der Spalte State wird gezeigt was der jeweilige Worker gerade macht. Hier steht entweder IDLE, falls der Worker nichts zu tun hat, oder der Name der Klasse des Objektes, welches er gerade verarbeitet. Im Normalfall lässt sich aus der Name der Klasse leicht ableiten um welche Art Statement es sich handelt.

Time

In der Spalte Time steht die Startzeit des letzten ausgeführten Statements, bzw. des Statements, das gerade ausgeführt wird.

24.2.5 Tab Sessions

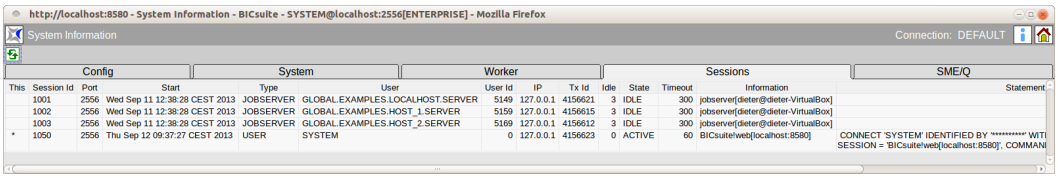


Abbildung 24.5: Session Information

Auf dem Tab ”Sessions” wird eine Tabelle aller momentan connected Sessions sowie ihre Aktivität gezeigt.

This

Die Spalte This zeigt mit einem Asterisk (*), welche der gezeigten Sessions die eigene Session ist.

Session Id

In der Spalte Session Id steht die Nummer der Session. Die Nummer der Session wird benötigt, wenn man mit dem ”KILL SESSION” Befehl die Session terminieren möchte.

Port

In der Spalte Port steht über welchen Port die Session mit dem Server connected ist.

Start

In der Spalte Start steht die Zeit zu der die Session aufgebaut wurde.

Type

In der Spalte Type steht um welche Art von Client es sich bei der Connection handelt. Die mögliche Werte sind USER, JOBSERVER oder JOB.

User

In der spalte User steht der Name des Benutzers. Im Falle eines Users ist es sein Benutzername. Wenn es sich um einen Jobserver handelt, ist es der vollqualifizierte Name des Jobservers. Bei einem Job wird die JobId gezeigt.

UserId

In der Spalte UserId steht die Objekt ID des Benutzers, Jobservers oder Jobs.

IP

In der Spalte IP steht die IP-Adresse vom Rechner von dem die Connection aufgebaut wurde.

TxId

In der Spalte TxId steht die Nummer der zuletzt durchgeführte Transaktion. Wenn gerade ein Statement ausgeführt wird, ist es die Nummer der aktiven Transaktion.

Idle

In der Spalte Idle steht die Anzahl Sekunden die nach der letzten Aktivität verstrichen sind. Idealerweise ist dieser Wert im Falle von Jobservern nicht höher als ihr NOP_DELAY.

State

In der Spalte State wird der Zustand der Session gezeigt.

Wert	**B**edeutung
IDLE	Die Session wartet auf Eingaben des Benutzers
QUEUED	Die Session hat ein Statement in der Warteschlange zum Ausführen bereit stehen
ACTIVE	Ein Statement der Session wird gerade ausgeführt
COMMITTING	Ein Statement der Session ist in der Abschlussphase der Ausführung
CONNECTED	Eine Session wurde initiiert aber der Benutzer hat sich noch nicht authentifiziert

Wird fortgesetzt

Fortsetzung der vorherigen Seite	
Wert	**B**edeutung

Tabelle 24.3: Session Status

Timeout

In der Spalte Timeout steht nach wie viele Sekunden Idletime die Connection von Server abgebrochen wird. Ein Wert von 0 bedeutet dabei, dass der Server die Connection nicht abbricht.

Statement

In der Spalte Statement steht das Statement, welches gerade von der Connection ausgeführt wird, so weit die Connection nicht IDLE ist, natürlich. Es ist selbstverständlich, dass Passwörter in Connect-Statements unerkennbar gemacht werden.
Allerdings ist das Statement nur dann sichtbar, wenn man über Administratorrechte verfügt.

24.2.6 Tab SME/Q

Auf dem Tab "SME/Q" wird eine Tabelle der je Quartal erzeugten Submitted Entities angezeigt. Der Tab "SME/Q" ist nur sichtbar falls der Benutzer der Gruppe "ADMIN" angehört.

Year

Die Spalte Year enthält das Jahr.

Quarter

Die Spalte Quarter enthält das Quartal.

Total

Die Spalte Total enthält die Gesamtanzahl an Submitted Entities, welche in diesem Quartal erzeugt wurden.

Expected Total

Die Spalte Expected Total enthält im aktuellen, noch nicht abgeschlossen Quartal (1. Zeile der Tabelle), die hochgerechnete zu erwartende Anzahl zum Quartalsende. Diese Zahl ist unverbindlich und dient ausschließlich der Orientierung. Für alle vergangenen Quartale ist das Expected Total gleich dem Total.

Avg. SME / Day

Die Spalte Avg. SME / Day enthält die durchschnittlich pro Tag erzeugte Anzahl an Submitted Entities.

25 Object Monitoring

25.1 Bild

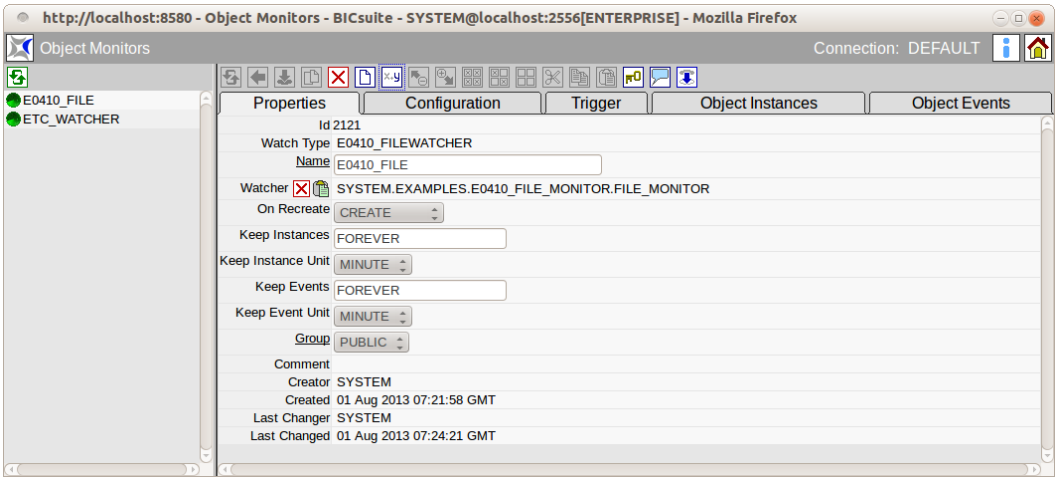


Abbildung 25.1: Object Monitoring

25.2 Konzept

25.2.1 Kurzbeschreibung

Der Dialog "Object Monitors" dient zum Verwalten von Objekt Monitoren. Ein Objekt Monitor überwacht die Existenz von Objekten und kann beim Erzeugen, Ändern oder Löschen von Objekten, Batches oder Jobs submitten.

25.2.2 Ausführliche Beschreibung

Ein Objekt Monitor kann die Existenz bzw. Eigenschaften beliebiger Objekte überwachen. Die für die Überwachung notwendigen Konfigurationsparameter, sowie die Eigenschaften der zu überwachenden Objekte, werden durch den, bei der Neuanlage eines Objekt Monitors gewählten "Watch Type" spezifiziert. "Watch Types" werden über das BICsuite Kommandozeileninterface (sdmsh) verwaltet.
Zur Überwachung wird ein sogenannter "Watcher" benötigt. Dieser muss als Job angelegt werden und sich konform dem im "Watch Type" des Objekt Monitors

festgelegten Interface verhalten. Der ”Watch Job” verbindet sich als ”Job” mit dem BICsuite Server und führt ein ”LIST OBJECT MONITOR” Kommando aus. Dieses Kommando liefert alle Objekt Monitoren welche den Job als ”Watcher” eingetragen haben. Für jeden gelieferten Objekt Monitor führt der ”Watcher” nun ein ”SHOW OBJECT MONITOR” Kommando aus, um die Konfigurationsparameter zu bestimmen, ermittelt alle der Spezifikation entsprechenden Objekte sowie deren Eigenschaften und liefert diese mittels dem ”ALTER OBJECT MONITOR” Kommando an den BICsuite Server zurück. Dabei wird vom ”Watcher” für jedes Objekt ein eindeutiger ”Unique Name” zur Identifizierung einer Objekt Entität generiert.

Die Screenshots dieses Kapitels basieren auf dem folgenden Beispiel ”Watch Type” zur Überwachung von Dateien:

```
create or alter watch type e0410_filewatcher
with parameters = (
     CONFIG DIRECTORY,
     CONFIG PATTERN,
     VALUE MTIME,
     INFO ATIME,
     INFO CTIME,
     INFO DEVID,
     INFO DIR SUBMIT,
     INFO FNAME SUBMIT,
     INFO GID,
     INFO INODE,
     INFO MODE,
     INFO SIZE,
     INFO UID
);
```

Dieser ”Watch Type” spezifiziert zwei Konfigurationsparameter (DIRECTORY und PATTERN), welche vom ”Watcher” Job ausgewertet werden, um die zu überwachenden Dateien zu identifizieren.

Der ”Watcher” Job muss für jedes gemeldete Objekt, die als VALUE oder INFO spezifizierten Objekteigenschaften liefern. Ändern sich Eigenschaften vom Typ ”VALUE” für eine Objekt Entität (Identischer ”Unique Name”) so wird ein ”CHANGE” Event ausgelöst. Eigenschaften vom Typ ”INFO” dienen nur informativen Zwecken und lösen keine ”CHANGE” Ereignisse aus.

Die mit der Option ”SUBMIT” versehenden Parameter werden beim Eintreten eines Ereignisses dem erzeugten ”Submit” bzw. ”Main” Job übergeben. INFO und VALUE Parameter werden nur dann übergeben, wenn je Objekt Entität ein Job erzeugt wird.

25.3 Navigator

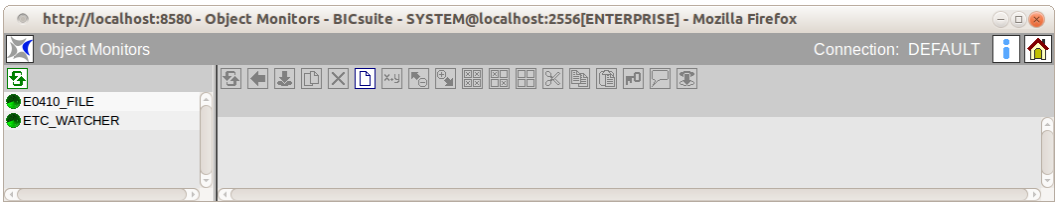


Abbildung 25.2: Object Monitoring; Navigator

Der Navigationsbildschirm des "Object Monitors" Dialoges zeigt die vorhandenen Objekt Monitore an.
Ist der angemeldete Benutzer Mitglied der Gruppe "ADMIN", so werden alle Objekt Monitore angezeigt. Ist dies nicht der Fall, so werden nur Objekt Monitore angezeigt, auf der aktuell angemeldete Benutzer Zugriffsrechte besitzt.

25.4 Editor

Bei der Neuanlage eines Objekt Monitors muss der "Watch Type" ausgewählt werden. Dieser kann später nicht mehr geändert werden.

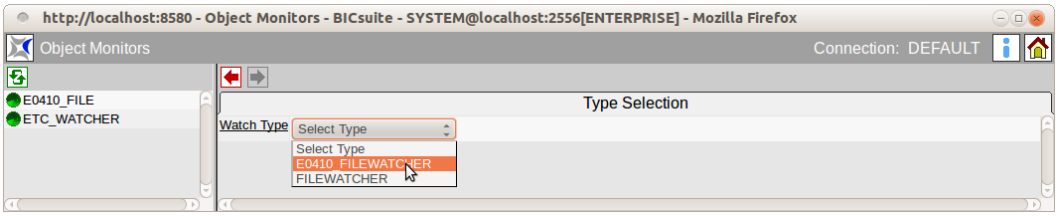


Abbildung 25.3: Object Monitoring Neuanlage

25.4.1 Tab Properties

Dieser Tab dient der Pflege der Eigenschaften von Objekt Monitoren.
Er sieht folgendermaßen aus:
Der Properties-Tab für Objekt Monitoren hat folgende Felder:

ID Die ID wird automatisch vergeben und dient zur eindeutigen systemweiten Identifikation des Objektes.

Watch Type Der bei der Neuanlage ausgewählte "Watch Type".

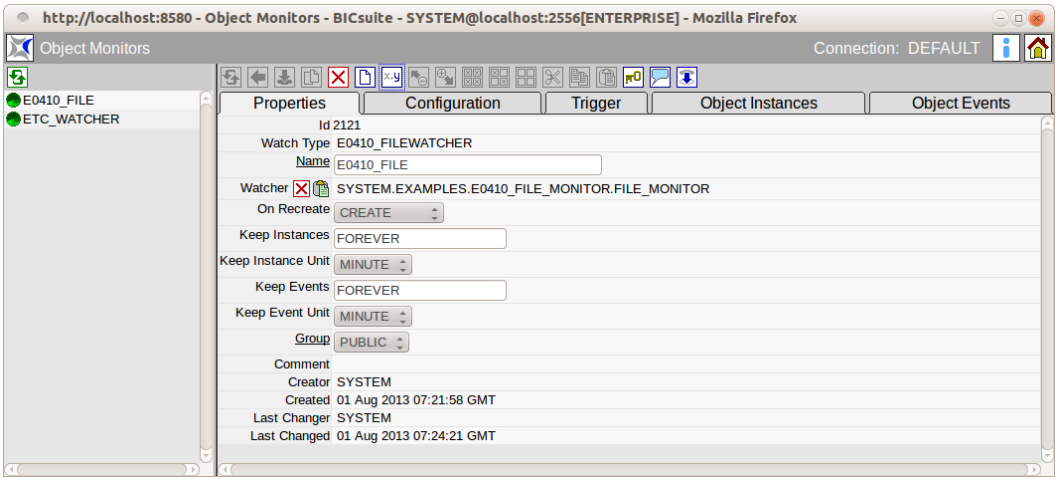


Abbildung 25.4: Object Monitoring Eigenschaften

Name Der Name des Objekt Monitors. Dieser ist frei wählbar und muss für alle Objekt Monitore eindeutig sein.

Watcher Hier wird der ”Watcher” Job eingetragen, der für die Informationsgewinnung des Objekt Monitors verantwortlich ist. Das Setzen eines ”Watchers” erfolgt durch *Copy* des ”Watcher” Jobs im ”Batches und Jobs” Dialog mit anschließendem Anklicken des *Paste* Icons im Label des Feldes. Durch durch Anklicken des ”Clear” Icons im Label des Feldes wird das Feld geleert.

On Recreate Im Feld ”On Recreate” wird eingetragen, wie das erneute Auftauchen einer Objekt Entität (nachdem diese gelöscht wurde) behandelt werden soll. Liste der Möglichkeiten für ”On Recreate”:

1. *CREATE*

   Das erneute Erzeugen löst ein ”CREATE” Ereignis aus.

2. *CHANGE*

   Das erneute Erzeugen löst ein ”CHANGE” Ereignis aus.

3. *DO NOTHING*

   Das erneute Erzeugen löst kein Ereignis aus.

Keep Instances Mit ”Keep Instances” und ”Keep Instance Unit” wird angegeben, wie lange Objekt Entitäten im System erhalten bleiben sollen, nachdem diese gelöscht wurden. Ältere Objekt Entitäten werden vom System nur entfernt, wenn der Objekt Entität keine Ereignisse mehr zugeordnet sind (Siehe ”Keep Events”).

Keep Instances Unit Gibt an, in welcher Einheit der Wert für ”Keep Instances” eingegeben wurde. Möglich sind MINUTE, HOUR, DAY, WEEK, MONTH und YEAR.

Keep Events Mit ”Keep Events” und ”Keep Events Unit” wird angegeben, wie lange Informationen über Events dieses Objekt Monitors im System erhalten bleiben sollen. Ältere Events werden nur entfernt, wenn die durch diese Events getriggerten Jobs im Status FINAL oder CANCELLED sind.

Keep Events Unit Gibt an, in welcher Einheit der Wert für ”Keep Events” eingegeben wurde. Möglich sind MINUTE, HOUR, DAY, WEEK, MONTH und YEAR.

Group Hier kann gewählt werden, welche Gruppe der Eigentümer des Objekt Monitors ist.

25.4.2 Tab Configuration

Dieser Tab dient der Konfiguration der Parameter für den ”Watcher” Job, welche im ”Watch Type” als ”CONFIG” spezifiziert wurden.
Er sieht folgendermaßen aus:

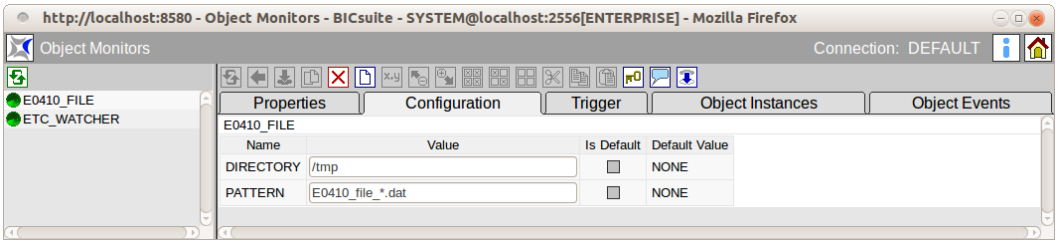


Abbildung 25.5: Object Monitoring Konfiguration

Der Configuration Tab enthält eine Tabelle mit folgenden Spalten:

Name Name des Konfigurationsparameters.

Value Wert des Konfigurationsparameters.

Is Default Hier kann gewählt werden, ob der eingegebene Wert oder der Default Wert des ”Watch Types” verwendet wird. Falls der ”Watch Type” keinen Default definiert, ist diese Checkbox nicht aktiv.

Default Value Der im ”Watch Type” spezifizierte Default Wert oder ”NONE”.

25.4.3 Tab Trigger

Dieser Tab zur Verwaltung der Trigger für dieses Objekt Monitor. Trigger lösen beim Eintreten eines Ereignisse (CREATE, CHANGE oder DELETE) den Submit von Batches oder Jobs aus und erzeugen dadurch Objekt Events.
Er sieht folgendermaßen aus:

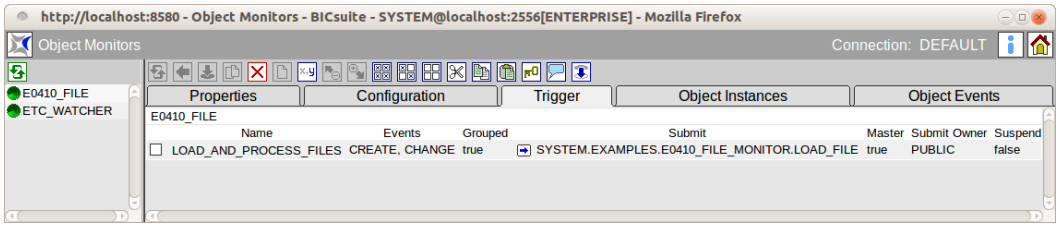


Abbildung 25.6: Object Monitoring Trigger

Die existierenden Trigger werden in einer Tabelle angezeigt.
Durch einen Mausklick auf den Trigger Namen, wird der Tab "Trigger Details" zur Bearbeitung der Trigger Eigenschaften geöffnet.
Für das Anlegen eines Triggers benutzt man die *Copy* (im Dialog "Batches und Jobs") und *Paste* Funktionalität der Benutzeroberfläche.
Die Tabellenspalten werden im folgenden Tab "Trigger Details" beschrieben.

25.4.4 Tab Trigger Details

Der Tab Trigger Details dient dem Bearbeiten der Eigenschaften eines Triggers.
Er sieht folgendermaßen aus:
Der Trigger Details Tab enthält folgenden Felder:

Trigger Name Name des Triggers. Dieser muss für einen Objekt Monitor eindeutig sein.

Active Checkbox mit der ein Trigger aktiviert bzw. deaktiviert werden kann. Deaktivierte Trigger erzeugen beim Eintreten eines Ereignisses keinen Event und lösen damit keinen Submit aus.

Comment Hier wird der Kommentar zu dem Trigger angezeigt falls dieser vorhanden ist.

Events to submit on Mit den drei Checkboxen für Create, Change und Delete wird ausgewählt, für welche Ereignisse ein Event erzeugt wird und damit ein Submit stattfinden soll.

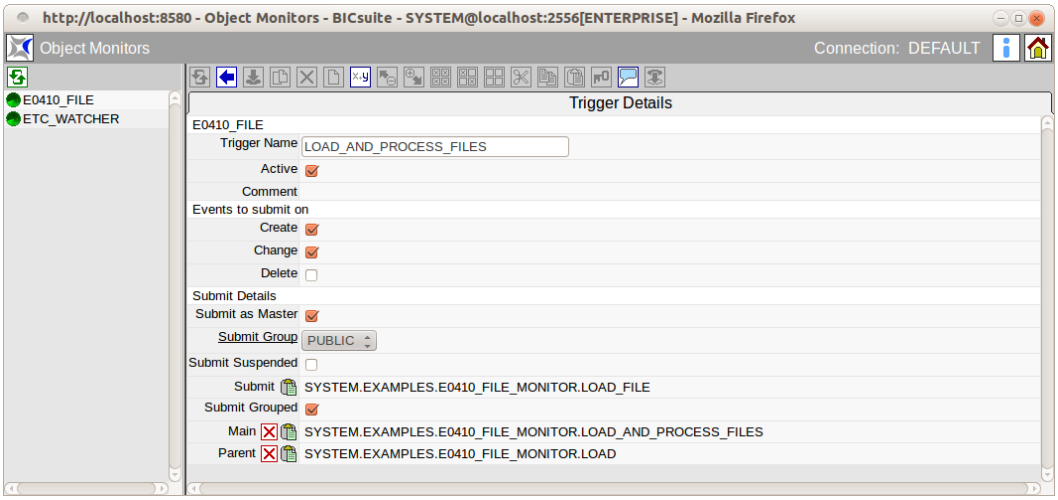


Abbildung 25.7: Object Monitoring Trigger Details

Submit as Master Ist diese Checkbox gesetzt, werden Batches oder Jobs beim Submit als Master erzeugt.
Ist sie nicht gesetzt, werden Batches oder Jobs beim Submit als Kinder des "Watcher" Jobs erzeugt. In diesem Fall muss der "Submit" bzw. "Main" Job (falls "Grouped Submit' gesetzt und "Main" spezifiziert wurde) als direktes Kind des "Watcher" Jobs definiert sein.

Submit Group Hier wird spezifiziert welche Gruppe der Eigentümer der beim Submit erzeugten Batches oder Jobs sein soll. Dieses Feld wird nur angezeigt wenn *Submit as Master* gesetzt ist.

Submit Suspended Ist diese Checkbox gesetzt, werden Batches oder Jobs beim Submit auf den Status SUSPENDED gesetzt und laufen nicht an. Ein Operator muss diese durch einen manuellen RESUME zur Ausführung freigeben.

Submit Definiert den Job, der beim Eintreten der ausgewählten Ereignisse erzeugt werden soll.
Ist das Feld *Submit Grouped* nicht gesetzt, wird je "ALTER OBJECT MONITOR" Befehl des "Watcher" Jobs, für jede Objekt Entität ein eigener Job erzeugt.
Ist das Feld *Submit Grouped* gesetzt und im Feld *Main* kein Job eingetragen, wird je "ALTER OBJECT MONITOR" Befehl des "Watcher" Jobs, für alle Objekt Entitäten nur ein Job erzeugt. Dieser ist dann selbst für die Verarbeitung aller betroffenen Objekt Entitäten verantwortlich.
Das Setzen eines "Submit" Jobs erfolgt durch *Copy* eines Batches oder Jobs im "Batches und Jobs" Dialog mit anschließendem Anklicken des *Paste* Icons im Label des

Feldes.
Das Feld ist vorbelegt mit dem Job, mit dem der Trigger durch ein *Paste* auf dem Trigger Tab angelegt wurde.

Submit Grouped Ist das Feld *Submit Grouped* gesetzt, werden Ereignisse wie folgt behandelt:
Ist im Feld *Main* kein Job eingetragen, wird je "ALTER OBJECT MONITOR" Befehl des Watcher Jobs für alle Objekt Entitäten der im Feld *Submit* angegebene Job nur einmal erzeugt.

Main Das Feld *Main* ist nur sichtbar falls das Feld *Submit Grouped* gesetzt wurde.
Ist im Feld *Main* ein Job eingetragen, so wird je "ALTER OBJECT MONITOR" Befehl des Watcher Jobs für alle Objekt Entitäten der im *Main* angegebene Job einmal erzeugt und für jede Objekt Entität ein eigener im Feld *Submit* angegebener Job erzeugt.
Falls im Feld *Parent* kein Job eingetragen wurde, so werden die erzeugten "Submit" Jobs als Kinder der "Main" Jobs angelegt. In diesem Fall muss der "Submit" Job als direktes Kind des "Main" Jobs definiert sein.
Das Setzen eines "Main" Jobs erfolgt durch *Copy* eines Batches oder Jobs im "Batches und Jobs" Dialog mit anschließendem Anklicken des *Paste* Icons im Label des Feldes. Durch durch Anklicken des *Clear* Icons im Label des Feldes wird das Feld geleert.

Parent Das Feld *Parent* ist nur sichtbar falls das Feld *Main* Job gesetzt ist.
Ist im Feld *Parent* ein Job eingetragen, so werden die erzeugten "Submit" Jobs als Kinder des "Parent" Jobs angelegt. In diesem Fall muss der "Parent" Job ein direktes oder indirektes Kind des "Main" Jobs sein und der "Submit" Job als direktes Kind des "Parent" Jobs definiert sein.
Das Setzen eines "Parent" Jobs erfolgt durch *Copy* eines Batches oder Jobs im "Batches und Jobs" Dialog mit anschließendem Anklicken des *Paste* Icons im Label des Feldes. Durch durch Anklicken des *Clear* Icons im Label des Feldes wird das Feld geleert.

25.4.5 Tab Object Instances

Des Tab "Object Instances" listet die dem System bekannten Objekt Entitäten und die, für diese erzeugten Events.
Er sieht folgendermaßen aus:
Je Objekt Entität wird als Icon ein farbiger Würfel, gefolgt vom eindeutigen Namen (Unique Name) angezeigt. Dieser ist grün falls das Objekt nie geändert bzw. gelöscht wurde, blau falls das Objekt modifiziert wurde und rot falls es nicht mehr existiert. Die Spalten "Created", "Changed" und "Removed" geben an, wann die

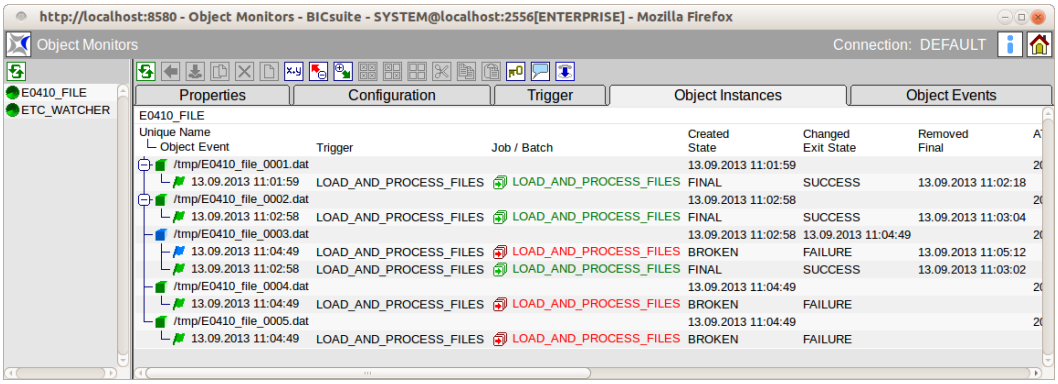


Abbildung 25.8: Object Monitoring Object Instances

Objekt Entität erzeugt, modifiziert und gelöscht wurde. Zusätzlich werden je Objekt dessen INFO und VALUE Werte angezeigt.
Wurden durch Ereignisse auf einer Objekt Entität Events erzeugt, so werden diese unter der entsprechenden Objekt Entität angezeigt. Diese können, wie in Baumansichten üblich, ein- und ausgeblendet werden. Gibt es für eine Objekt Entität Events, deren Job State weder FINAL oder CANCELLED ist, so werden die Events dieser Objekt Entität immer eingeblendet und können auch nicht ausgeblendet werden.
Für Events wird als Icon eine Fahne angezeigt (Grün, Blau oder Rot für CREATE, CHANGE und REMOVE Events), wann der Event ausgelöst wurde und welcher Trigger dafür verantwortlich war. Weiterhin wird der erzeugte ”Submit” bzw. ”Main” (falls ”Grouped Submit” und ”Main” im Trigger gesetzt wurde) Job mit den Informationen über seinen Ausführungsstatus angezeigt. Durch einen Klick auf den Job Namen wird ein Monitor Fenster für diesen Job geöffnet.

25.4.6 Tab Object Events

Des Tab ”Object Events” listet die vom Objekt Monitor erzeugten Events.
Er sieht folgendermaßen aus:
Für jeden durch den Objekt Monitor erzeugte Event wird als Icon eine Fahne angezeigt (Grün, Blau oder Rot für CREATE, CHANGE und REMOVE Events), wann der Event ausgelöst wurde und falls ”Grouped Submit” nicht gesetzt wurde, welcher Trigger dafür verantwortlich war.
Weiterhin wird,falls ”Grouped Submit” nicht gesetzt oder ”Main” im Trigger gesetzt wurde , der erzeugte ”Submit” Job mit den Informationen über seinen Ausführungsstatus angezeigt. Der Unique Name der Event erzeugenden Objekt Entität wird ebenfalls angezeigt. Durch einen Klick auf den Job Namen wird ein Monitor Fenster für diesen Job geöffnet.
Falls ”Grouped Submit” gesetzt war, werden die Events für einzelne Objekt En-

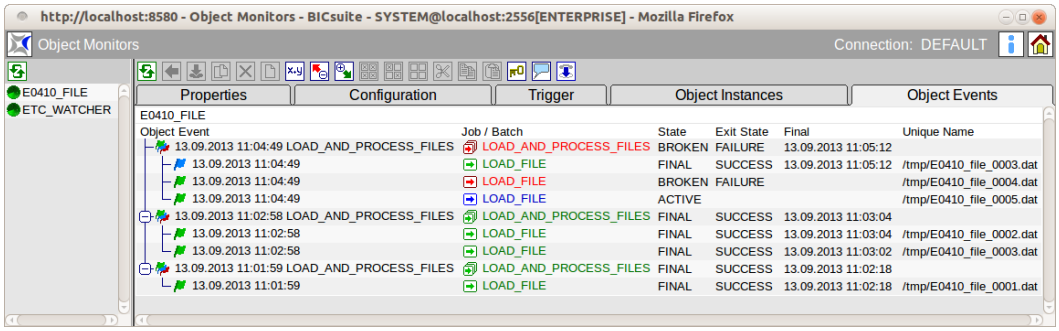


Abbildung 25.9: Object Monitoring Object Events

titäten unter einer Gruppen Event Zeile zusammengefasst. Die Gruppen Event Zeilen werden durch ein Icon mit mehreren Fähnchen, gefolgt von Zeitpunkt des Events und dem ”Main” bzw. ”Submit” Job (falls kein ”Main” Job definiert war) mit den Informationen über seinen Ausführungsstatus dargestellt.
Die Events unter Gruppen Events können, wie in Baumansichten üblich, ein- und ausgeblendet werden. Gibt es für einen Gruppen Event, einen Event dessen Job State weder FINAL oder CANCELLED ist, so werden die Events dieses Gruppen Events immer eingeblendet und können auch nicht ausgeblendet werden.

Index

Abhängigkeit, 40, 41, 130
Ablaufumgebung, 69, 110
Active distribution, 98
Active Schedule, 177
Administrator, 30
After Final, 154
Alias, 124
Allocated, 87
Amount, 65, 74, 85, 87, 111, 138, 141
Anzahl Sessions, 250
Audit, 197, 230
Ausführungsort, 70
Ausführungsumgebung, 119
Ausfallzeit, 174
Auto-Refresh, 194
Automatischer Restart, 122
Autoresume, 210
Autostart, 205
Available, 87

Batch, 103, 105, 117, 185, 237
Batch Default, 41
Batch Konvention, 105
Before Final, 154
Benutzer, 2, 165, 166, 169
Benutzergruppe, 69, 109, 116
Benutzerkennung, 2
Benutzername, 2
Benutzerverwaltung, 161
Betriebssystem, 251
Blocked, 87
Bookmark, 106, 189, 193, 204, 233
Bourne-Shell, 119
Broken Active, 199
Broken Finished, 199
Broken Flag, 41
Broken State, 41
Browser, 1, 2

Calendar, 183
Calendar horizon, 243
Cancel, 211
Cancel Button, 10
Cancelled, 198
Chain, 160
Child-Prozess, 117
Childreference, 145
Children, 123
Clear Warning, 211
Clone Button, 11
Collapse All, 7
Comment, 31
Condition, 71, 132, 137, 203
Connection, 255
Constant, 63, 85
Copy Button, 14
Create Recht, 29
Current Amount, 84, 97
Cut Button, 14

Datenübergabe, 104
Datenbanksystem, 243
Day of Month Intervall, 180
Day of Week Intervall, 180
Default Directory, 120
Default Environment, 164
Defined Amount, 84, 97
Defined Resource, 141
Definitionsfehler, 198
Dependency, 41, 123, 130, 159
Dependency Default, 41
Dependency Mode, 130

Dependency Wait, 198
Dependent Job, 160
Deselect All Button, 13
Desktop, 6
Detail Bookmark, 190
Distribution, 97–99
Drop Button, 11
Drop line, 18
Drop Recht, 28
Duplizieren, 11, 109
Dynamic Children, 124
Dynamische Priorität, 215

Edit Button, 15
Edit Recht, 28
Effective Horizon, 177
Effektive Priorität, 89
Eigentümer, 109, 116
Enabled, 81
Environment, 28, 67, 69, 106, 110, 119, 135, 136, 164, 217
Environment-Variablen, 119
Error, 41, 195, 198
Error Logfile, 121, 217
Evaluation Cycle, 97
Exclusive, 88
Execute Recht, 29
Exit Code, 35, 215
Exit Code Ranges, 36
Exit Find, 8
Exit State, 33, 35, 39, 43, 51, 104, 109, 197, 202, 214
Exit State Definition, 33
Exit State Mapping, 35, 40, 119, 215
Exit State Profile, 39, 42, 43, 109, 119, 214
Exit State Translation, 42, 44, 128
Expand All, 8
Expected Finaltime, 120, 148
Expected Runtime, 120, 217
Expiration, 66, 140
Expire, 66, 140
Export, 15, 55
Export Mode, 56
Export Script, 56
Expression, 143, 145

Factor, 85
Fehler-Status, 42
Filterkriterien, 193, 200
Final, 199
Final State, 40
Finish Child, 155
Finished, 199
Fire Limit, 156, 250
Folder, 28, 59, 69, 106
Folder Bookmark, 189
Folder Environment, 69, 70, 110
Folder Konvention, 109
Footprint, 67, 73, 120, 135, 136, 217
Free Amount, 65, 84, 97

Garbage Collection, 252
Grant, 171
Grant Button, 15
Group, 28
Gruppe, 33, 36, 40, 44, 47, 52, 59, 60, 70, 74, 79, 97, 109, 166, 169, 261
Gruppen Event, 267

Hauptmenü, 2
Help Icon, 6
Hide Dependencies, 159
Hide Folderpath Button, 11
Hide Hierarchypath Button, 12
Hide Leaves, 8
Hide Locked, 8
Hierarchy, 8
Historie, 202, 244, 245
Home Icon, 6
Horizont, 177

Idle time, 256
Ignored, 87
Ignored Dependency, 128
Immediate Local, 153
Immediate Merge, 153

Import, 55
Import/Export Button, 15
Intervall, 178
IP-Adresse, 255
ISO Kalenderwoche, 182
ISO Norm, 117
ISO Week of Month Intervall, 181
ISO Week of Year Intervall, 182
ISO Woche, 181

Java Virtual Machine, 251
JDBC Driver, 245
JdbcDriver, 253
Job Definition, 28, 40, 65, 69, 70, 73, 103
Job Environment, 70
Job Server, 70
Job State, 202, 207, 213
Job Status, 122, 194, 197
Jobserver, 28, 77, 198, 199, 216
Jobserverprozess, 81

Kalender, 120, 177, 183, 202, 238, 243
Kalender Horizont, 177
Kalenderfunktion, 237
Kalenderhorizont, 202
Kalenderwoche, 182
Kategorie, 60, 62
Keep, 66, 75, 88, 138
Kill, 211
Kill Program, 120, 147, 199, 211, 217
Killed, 120, 199
Kommandosprache, 55
Kommentar, 31, 208, 211
Kompatibilitätsmatrix, 88
Konfiguration, 242
Konfigurationsparameter, 93
Konvention, 105
Kopfzeile, 2

Löschen, 11
Löschen einer Zeile, 18
Lastverteilung, 104, 139
Laufzeitinformation, 208
Laufzeitumgebung, 69
Load Tracing, 90
Local Constant, 63, 85
Lockmode, 66, 87, 88, 140
Logfile, 121, 216
Logfile Pattern, 95
Logfile Write, 121
Logging, 249
Login-Maske, 2

Main Desktop, 2, 6
Main Schedule, 174
Manage Privileges, 30
Manage Privileges Tab, 170
Manage Privilegien, 167, 170
Master Bookmark, 190
Master Job, 103, 117, 128, 138, 139, 156, 185, 190, 193, 200, 214
Master Reservation, 87
Memory, 252
Merge Global, 128
Merge Local, 127
Merge Mode, 126, 127, 214
Milestone, 103
Minimale effektive Priorität, 118
Mode Search, 190
Monatsdarstellung, 237
Monitor Recht, 29
Month of Year Intervall, 182

Name Pattern, 202
Named Resource, 28, 59, 60, 66, 71, 74, 75, 85
New Button, 11
Nice Value, 117, 125, 213
NoLock, 88

Objekt Monitor, 259
Operate, 28
Operate Recht, 29
Optionsfeld, 2
Ordner, 8
Owner, 59, 79, 97, 153, 213

Parameter, 63, 85, 91, 104, 106, 110, 119, 142, 185, 203, 242
Parameter Type, 143
Parent, 42, 43
Passwortabfrage, 2
Paste Button, 14
PENDING State, 40
PID, 81, 147
Pinning, 106
Pool, 62, 66, 245
Pooled Resource, 96
Priorität, 42, 89, 117, 118, 125, 215, 246
Priority Aging, 118
Privileg, 33, 36, 40, 44, 47, 52, 70, 74, 166, 169
Process Id, 120
Programm, 69

Query Maske, 200, 206, 234
Quoting, 119

Range of Day Intervall, 179
Rechte, 59, 169
Reference, 144
Refresh Button, 7
Regular Expression, 95
Regular expression, 202
Repeat Intervall, 179
Requestable Amount, 65, 83, 85, 111, 141
Requested, 86
Required Resource, 137
Requirement, 137
Rerun, 196, 210
Rerun Children, 197, 210
Rerun Program, 120, 216
Reservation, 87
Reserved, 87
Resource, 28, 65
Resource Anforderungen, 69
Resource Definition, 60
Resource Factor, 85
Resource Link, 101
Resource Pool, 62, 66
Resource Recht, 29
Resource State Definition, 45
Resource State Mapping, 51, 66, 89, 140
Resource State Profile, 47, 62, 83, 84, 111, 140
Resource Status, 45, 47, 51, 62, 66, 84, 140
Resource Trigger, 63
Resource Wait, 198
Resource-Scheduler, 247
Resource-Status, 89, 111
Resourcereference, 145
Ressource, 69, 73, 77, 81, 104, 110, 135
Ressource Standard Parameter, 64
Restart, 122
Restartable, 41
Resume, 126
Resume Zeitpunkt, 117, 126, 157, 188, 209
Run Program, 120, 198, 216
RunMode, 247
Runnable, 198
Running, 199
Running Jobs, 233
Running Master Jobs, 193
Runtime, 197

Save View, 8
Scheduling Entity, 64, 103
Scope, 28, 59, 65, 77
Script, 56
Search, 8
Search Bookmark, 190
Select All Button, 13
Select Button, 15
Select Statement, 247
Server Connection, 191
Server User, 165
Server Verbindung, 2
Serverbenutzer, 161
Serverkonfiguration, 242
Serverparameter, 243
Session, 254

Session Timeout, 247
Sessions, 250
Set State, 211
Shared, 88
Shared Compatible, 88
Shared Exclusive, 88
Show Folderpath Button, 11
Show Hierarchypath Button, 12
Show Leaves, 8
Show Locked, 8
Speicher, 252
Speichern, 10
Speichern Button, 11
SQL LIKE Syntax, 108
Standard Button, 10
Standard Parameter, 120, 146
Standardparameter, 119
Standardvariablen, 244
Start Find, 8
Start Mode, 205
Started, 199
Starting, 198
Startzeit, 238
Startzeiten, 174
Static Ressource, 62
Statusübergang, 199
Sticky, 66, 87, 88, 138
Sticky Name, 140
Sticky Parent, 140
Sub Schedule, 174
Submit, 28, 185
Submit Recht, 29
Submit Suspended, 125
Submitted, 198
Submitted Entity, 39, 59
Suchmuster, 108
Suspend, 185
Suspended, 81, 196
Synchronize Wait, 198
Synchronizing Resource, 45, 47, 62, 104, 138
System Bookmark, 107
System Bookmarks, 190
System Resource, 62, 66, 73
Systemobjekten, 30
Systemprivilegien, 30
Systemvariable, 120

Tagesdarstellung, 238
Target Amount, 97
TCP-Port, 246
Threads, 243, 250, 251
Tickets, 248
Time of Day Intervall, 179
Time Scheduling, 15, 173, 237
Timeout, 135, 256
Timeout State, 136
Timer-Thread, 248
Timestamp, 217
To Kill, 199
Toggle Selection, 13
Toggle Selection Button, 13
Trace Level, 249
Trigger, 41, 63, 104, 149, 197, 250, 264
Trigger type, 153
Trigger-Parameter, 149

Umbenennen, 109
Umgebungsvariable, 91, 94
Unique Name, 267
Unreachable, 41, 134, 195, 198
Unreachable State, 41
Unresolved Dependency, 213
Unresolved Handling, 132
Until Final, 155
Until Finished, 155
Up Button, 11
Usage, 62
Use Recht, 28
User, 165, 169
User Bookmark, 190

Variable, 199
Vererbung, 92
View Privileg, 169
View Recht, 28
Virtual Machine, 251

Vorgänger, 41

Warning Trigger, 156
Warnung, 42, 156, 197
Watch Type, 261
Watcher, 266
Week of Month Intervall, 181
Windows-Zwischenablage, 20
Wochendarstellung, 237
Wochentag, 181
Worker Threads, 253
Working Directory, 120, 121

Zeitpläne, 173
Zufügen einer Zeile, 17
Zugangskontrolle, 165
Zugriffsmodus, 140
Zugriffsrechte, 15, 28, 56, 166, 261
Zwischenablage, 14, 20