



independIT Integrative Technologies GmbH

BICsuite New Features in Version 2.4

14. Dezember 2007

Copyright © 2007 independIT GmbH

Rechtlicher Hinweis

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Dokumentes, 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.

Übersicht der New Features

In der Version 2.4 der BICsuite gibt es abgesehen von Bugfixes und internen Verbesserungen, eine Reihe von neuen Features. Die wichtigsten dieser Features werden in der untenstehenden Übersicht in Kurzform aufgelistet. Im folgenden Abschnitt wird zu den einzelnen Änderungen eine detaillierte Information gegeben.

- Verbesserte Dokumentationsmöglichkeiten
- Neue Systemvariablen
- Conditional Dependencies
- Vereinfachtes Operating
- Recursive Variable Resolution
- Lokale Variablen für Batches und Jobs
- Rekursive Grants
- Vereinfachte Definition von Trigger
- Pooled Resources
- Warning Triggers
- Warnung wenn Trigger Limit erreicht wird
- Negative Expirations
- Jobserver Notification
- Copy Scope im Frontend
- Resource Tracing
- Resource Factor zum Skalieren von Requirements
- Dump Command und Import/Export aus der Oberfläche
- Edit Children im Hierarchy-View
- Deregister Jobserver
- Conditional Resource Requirements for static Resources

Detaillierte Information

Verbesserte Dokumentationsmöglichkeiten

Folgende Objekttypen können nun mit einem Kommentar versehen werden:

Environment	Event	Exit State Definition
Exit State Profile	Exit State Mapping	Exit State Translation
Folder	Footprint	Group
Interval	Scheduling Entity	Named Resource
Parameter	Resource	Resource State Definition
Resource State Profile	Resource State Mapping	Schedule
Scheduled Event	Scope	Trigger
User		

In der Buttonleiste der Oberfläche wurde dazu ein neuer Button () aufgenommen. Nach Betätigung dieses Buttons wird ein neues Fenster zur Eingabe des Kommentars geöffnet.

Neue Systemvariablen

Für Jobs stehen nun auch folgende standard Parameter zur Verfügung:

Name	Bedeutung
WORKDIR	Der Pfad des aktuellen Working Directories
WARNING	Es wurde die Systemvariable WARNING eingeführt. Diese Variable enthält den Text des letzten Auditeintrages der das Hochsetzen des Warncounts zur Folge hatte. Dies allerdings nur dann, wenn der Warncount > 0 ist.
RERUNSEQ	Diese Variable gibt an wie oft der Job bereits restartet wurde.
SCOPENAME	Liefert den vollen Pfadnamen des Jobserver, der den Job ausführt bzw ausgeführt hat.
TRIGGERWARNING	Diese Variable ist nur gültig im Kontext eines Warning-Triggers. Sie enthält die Meldung die beim Auslösen der Warning ausgegeben wurde.

Conditional Dependencies

Bei der Definition von Dependencies ist es in der Enterprise Version von BICsuite möglich eine zusätzliche Bedingung zu definieren. Die Dependency gilt nur dann als erfüllt, wenn die spezifizierte Bedingung wahr ergibt. Diese Bedingung kann generell für die gesamte Dependency spezifiziert werden, oder aber einzeln für jeden geforderten State.

Sind sowohl eine Bedingung für die gesamte Abhängigkeit als auch eine Bedingung für einen bestimmten Status spezifiziert, müssen beide Bedingungen erfüllt sein um die Abhängigkeit zu genügen.

Die Syntax für die Bedingungen folgt den Regeln der Bedingungen für conditional Triggers.

Vereinfachtes Operating

Es ist jetzt möglich direkt aus der Jobliste heraus zusätzlich die Operatoraktionen Resume () , Display Logfiles () , Rerun () und Rerun Children () auszuführen.

Natürlich ist nach wie vor die Möglichkeit des Cancel () gegeben. Die entsprechenden Buttons werden, wenn die Aktion möglich ist, in der Jobliste hinter dem Job angezeigt. Der Resume-Button formt dabei eine Ausnahme und wird vor dem Job angezeigt.

Recursive Variable Resolution

Variablen werden nun rekursiv ausgewertet. Das bedeutet, wenn innerhalb einer Variable etwas in der Form \$NAME bzw. \${NAME} auftritt, wird die Variable NAME aufgelöst und den 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 \\. Wichtig ist, dass die Variable jeweils im Kontext des definierenden Jobs ausgewertet wird. Wenn also ein Job A einen Import-Parameter X deklariert und sein Parent diese Variable mit Inhalt \$Y anbietet, dann wird die Variable Y im Kontext des Parents ausgewertet.

Lokale Variablen für Batches und Jobs

Es gibt nun die Möglichkeit lokale Variablen zu definieren. Diese Variablen sind nur aus der Sicht des definierenden Jobs sichtbar. Wenn ein Job A einen Import-Parameter X deklariert und sein Parent eine lokale Variable X definiert, ist diese für den Job A nicht sichtbar.

Rekursive Grants

Der Grant bzw. Revoke kann nun auch "rekursiv" durchgeführt werden. Dies hat drei Dinge zur Folge:

1. "unbekannte" Privilegien werden ignoriert.
2. Grants für Objekte auf die man keine Grant-Rechte hat werden (lautlos) ignoriert. Die Rekursion hört an dieser Stelle auf.
3. Falls ein Objekt Kinder hat, wird der Grant auch für diese Kinder durchgeführt.

Desweiteren sind nun auch Grants für "children only" möglich. Dazu sind keine Privilegien auf dem Parent-Objekt notwendig. Allerdings benimmt sich ein nicht rekursiver Grant wie ein mehrfacher Grant innerhalb eines Multicommands. Das heisst: geht auch nur einer der Grants schief, wird keiner der Grants durchgeführt. Um Fehler zu ignorieren kann jetzt die Force-Option genutzt werden. Die Cascade-Option beinhaltet die Force-Option also implizit.

Neu ist ebenfalls, dass mehrere Privilegien, sowie mehrere Gruppen auf einmal pro Grant Statement spezifiziert werden können.

Syntax:

```
GRANT|REVOKE privileg {, privileg} ON [CHILDREN OF] objecturl  
TO|FROM group {, group} [CASCADE|FORCE]
```

Aus der Oberfläche wird der Grant-Dialog wie vorher mit dem Grant-Button () aufgerufen. Falls dies aus einem hierarchischen Kontext erfolgt (Folder, Named Resource, Scope), gibt es auf der geöffneten Maske die entsprechenden Möglichkeiten Grants rekursiv zu erteilen.

Vereinfachte Definition von Trigger

Trigger benötigen jetzt nicht mehr zwingend eine Child Definition (Ausnahme AFTER.FINAL Trigger). Wird beim non master Submit eines Triggers keine Child Definition gefunden, so wird eine default Child Definition verwendet. Existiert eine Child Definition, so wird diese auch verwendet.

Pooled Resources

Zentraler Bestandteil der Enhanced Resource Management Funktionen der Enterprise Version sind die Resource Pools. Ein Resource Pool ist eine Menge von System Resources (oder Resource Pools) die zusammen einen zentral verwalteten Amount haben. Dieser Amount wird konform den im Resource Pool festgelegten Regeln über die teilnehmenden Resources und Pools verteilt. Im Falle von Resource Pools wird der zur Verfügung gestellte Amount wiederum weiterverteilt an den dem Pool angehörenden Resources.

Die Verteilung von Amounts erfolgt im Wesentlichen zweistufig. Periodisch werden von einem unabhängigen Thread sogenannte Target Amounts für alle Ressourcen und Pools eines Pools festgelegt. Diese Target Amounts stellen anzustrebende Werte dar. Hat eine Resource einen höheren Amount als ihren Target Amount, so wird sie bei Freigaben ihren Amount schnellst möglich dem Target Amount anpassen. Gibt es Resourceanforderungen die eine Resource nicht aus ihrem Amount befriedigen kann, so wird sie beim Pool mehr anfordern. Diese Anforderungen werden honoriert wenn ausreichender Amount zur Verfügung steht.

Wenn die Target Amounts erreicht sind und weiterhin Amounts zur Verfügung stehen, ist es den Ressourcen möglich über ihren Target Amounts hinaus weitere Amounts anzufordern (allerdings nur bis zum im Pool definierten Maximum).

Warning Triggers

Diese Triggers werden gefeuert wenn der Warning-Flag für ein Job gesetzt wird. In der Systemvariable TRIGGERWARNING steht der begleitende Text. Natürlich kann der Trigger noch mittels einer Condition spezifischer auf die vorliegende Situation reagieren.

Warnung wenn Trigger Limit erreicht wird

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

Negative Expirations

Ein negativer Wert beim Resource Expiration hat eine Semantik bekommen. Ursprünglich war es eine nicht erfüllbare Bedingung denn die Resource würde schon in der Zukunft als "zu alt" bewertet. Nun hat eine negative Expiration zur Folge, dass eine Resource mindestens so "alt" als angegeben sein muss.

Jobserver Notification

Jobserver hören, wenn konfiguriert, auf einen UDP-Port und können dadurch geweckt werden. Dies erhöht die Responsivität und reduziert die Systemlast. Der dazu verwendete Konfigurationsparameter, der in der Oberfläche auf dem "Config"-Tab aufgeführt wird, heisst NOTIFYPORT. Wenn für diesen Parameter keinen Wert spezifiziert wird, verhalten sich die Jobserver wie gehabt.

Copy Scope im Frontend

Copy Scope kann jetzt mittels copy and paste auf dem Properties Tab durchgeführt werden. Der zu kopierenden Jobserver (oder Scope) wird im Navigator selektiert und im Editor wird der Properties Tab selektiert. Dann wird der Copy-Button betätigt. Anschliessend wird im Navigator der Zielscope selektiert und mittels des Einfügen-Buttons wird eine Kopie erstellt.

Resource Tracing

Es ist nun möglich die Belegung von Ressourcen sowie Pools zu protokollieren. Die Auslastung einer Resource wird, wenn entsprechend angefordert, periodisch in die Tabelle RESOURCE_TRACE geschrieben. Dieses Feature gehört zu den Enhanced Resource Management Funktionen der Enterprise Version.

Um die Trace Tabelle leichter auswerten zu können, können Resources und Pools nun mit einem Tag versehen werden. Dieses Tag soll innerhalb Resources und Pools eindeutig sein. (D.h. auch das Benutzen eines Tags für sowohl eine Resource als auch einen Pool ist verboten). Das Tag ist auch Bestandteil der RESOURCE_TRACE Tabelle.

Das Tag wird in der Syntax als String spezifiziert und unterliegt somit nicht den Anforderungen der üblichen Identifier.

Resource Factor zum Skalieren von Requirements

Um Resource Anforderungen von Jobs von aussen justieren zu können, wurde als Feature der Enterprise Version 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

$$\text{ceil}(\text{Anforderung} * \text{Factor})$$

verwendet.

Dump Command und Import/Export aus der Oberfläche

In der Enterprise Version von BICsuite gibt es jetzt vielfältige Möglichkeiten zur Sicherung bzw. Exportieren von Objektdefinitionen ins Filesystem. Die dabei erzeugten Dateien enthalten BICsuite-Befehle die zur Wiederherstellung (Import) eingelesen werden können. Die Dump Funktionalität erlaubt die Sicherung von Einzelobjekten bis hin zu komplette zusammenhängende Strukturen.

Mit Hilfe des Import/Export-Buttons ( und ) ist die Export- und Importfunktionalität auch aus der Oberfläche abrufbar.

Der Export im Frontend ist customizable, so dass für unterschiedliche Anforderungen entsprechende Export-Arten bereit gestellt werden können.

Edit Children im Hierarchy-View

Im Hierarchy-View Fenster können jetzt nicht nur die Abhängigkeiten (Dependencies) sondern auch die Parent-Child Struktur bearbeitet werden. Dazu gibt es nun die Buttons Add Children () und Remove Child (). Beim "Add Children" wird zuerst ein Job, Batch oder Milestone selektiert. Daraufhin öffnet ein Fenster mit der vorhandenen Folderstruktur, aufgeklappt bei dem vorhin gewählten Objekt. Es können nun weitere Jobs, Batches und/oder Milestones selektiert und in die Zwischenablage kopiert werden. Im Hierarchy-View Fenster wird nun das Parent-Objekt selektiert und die in der Zwischenablage kopierten Jobs, Batches und/oder Milestones werden als Children zugefügt.

Deregister Jobserver

Mit Hilfe des Deregister-Buttons () können Jobserver nun aus der BICsuite!web Oberfläche deregistered werden. Der Jobserver bekommt nach dem Deregister keine Jobs mehr zugeteilt und laufende Jobs werden in den Status BROKEN_FINISHED versetzt. Diese Aktion wird benutzt um Jobserver, von denen bekannt ist, dass sie längere Zeit nicht mehr zur Verfügung stehen werden, zu deaktivieren.

Conditional Resource Requirements for static Resources

Eine Static Resource mit Condition kann nur dann belegt werden, wenn die Condition true ergibt. Dies wird bei dem Übergang von DEPENDENCY_WAIT nach SYNCHRONIZE_WAIT (und damit implizit auch beim Reschedule) geprüft. Das Setzen eines Parameterwertes einer Static Resource führt nun zum Reschedule. Dieses Feature der Enterprise

Version erlaubt einen parametergesteuerten dynamischen Auswahl der Laufzeitumgebung einzelner Jobs.