
Workflow Event Server

Benutzerhandbuch

Version 1.1

SerCon GmbH

Service-Konzepte für Informations-
Systeme GmbH

Geschäftsstelle Mainz

Heinrich - von - Brentano Str. 2

55130 Mainz

06131-978-0

06131-978-111



Stand:

09.10.2002 5:18

Autor:

Franziska Gaudeck

Inhaltsverzeichnis

1	Einleitung	3
1.1.....	Anwendungsbeispiel	3
2	Installation	5
2.1.....	Voraussetzungen	5
2.2.....	Installation	5
2.3.....	Starten des Workflow Event Server	6
2.4.....	Konfiguration	6
3	Modellieren in MQSeries Workflow	9
3.1.....	Workflow Konfiguration	9
3.2.....	Workflow Design	12
4	Kontrollmöglichkeiten durch den Administrator	21
4.1.....	Administration im Client	21
4.2.....	Kommandos für die Konsole	23
5	File Event	26
5.1.....	Funktionsweise	26
5.2.....	Parameterübersicht	27
5.3.....	Nicht kombinierbare Parameter	27
5.4.....	Beispiele	28
6	Time Event	29
6.1.....	Funktionsweise	29
6.2.....	Parameterübersicht	29
6.3.....	Beispiele	30
7	Hinweise	31
8	Beispielszenario	32
8.1.....	Konzeptübersicht	32
8.2.....	Beschreibung der Simulation	33
9	Anhang	39
9.1.....	Benötigte Software	39
9.2.....	Literatur	39

1 Einleitung

Der Workflow Event Server ist eine Erweiterung des IBM Produktes MQSeries Workflow. MQSeries Workflow ist ein elektronisches Werkzeug zum Abbilden und Automatisieren von Geschäftsprozessen. Ein Geschäftsprozess beschreibt einen internen Vorgang oder einen internen Vorgang mit externer Anbindung. Dieser Vorgang kann zum Beispiel die Handhabung der Verwaltung, der Einkauf von Materialien oder die Formalitäten für eine Ausschreibung umfassen. Ein Vorgang kann aus Teilprozessen bestehen, die im engeren Sinne auch Vorgänge sind, wie zum Beispiel die Abrechnung der Reisekosten, die Archivierung von Informationen oder das Ausfüllen von Aufträgen. Elektronische Werkzeuge wie MQSeries Workflow unterstützen das Management beim Entwerfen, Abbilden, Dokumentieren und Automatisieren von Geschäftsprozessen.

Der Workflow Event Server erweitert die Möglichkeiten der Prozessabarbeitung in MQSeries Workflow. Er umfasst die Events *File* und *Time*. *File* bietet die Option ein Verzeichnis auf das Eintreffen einer oder mehrerer Dateien zu überwachen, Dateien zu kopieren, verschieben oder zu löschen. Damit wird ermöglicht, den Prozessfluss durch das Eintreffen oder Vorhandensein von Dateien zu beeinflussen. *Time* ermöglicht den Prozess nach einer bestimmten Zeit, zu einem bestimmten Zeitpunkt oder Wochentag fortzusetzen. Hier handelt es sich um eine Zeitfunktion, die Steuerung und Überwachung des Prozesses erleichtern.

1.1 Anwendungsbeispiel

Mit MQSeries Workflow kann man leicht immer wiederkehrende Vorgänge als Verknüpfung von Aktivitäten darstellen. Dem hier gewählten Beispiel (siehe „*Abbildung 1: Beispiel für einen Prozess mit File und Time Events*“ Seite 4) liegt der Prozess „Ausschreibung“ zugrunde.

Bei einer Ausschreibung werden zuerst die Informationen aufgenommen, dann firmenintern bearbeitet. Erst wenn dies vollständig erfolgt ist, kann das Angebot erstellt werden. Durch die Nutzung des File Event Servers ist es möglich, auf das Eintreffen aller Dateien, die für die Angebotserstellung nötig sind, zu warten. Nach der Auswertung können verschiedene Fälle eintreten.

Im ersten dargestellten Fall wird die Weiterleitung erst zu einem vorbestimmten Zeitpunkt (Termin) fortgesetzt, dies wird mit dem Time Event realisiert. Im zweiten Fall wird das Angebot zur abschließenden Durchsicht sofort weitergeleitet. Der dritte Weg beschreibt den Fall der Wiedervorlage. Die Wichtigkeit und die Komplexität des Projektes entscheidet, welcher Fall zutreffend ist. Bevor aber das fertige Angebot dem Kunden vorgelegt werden kann, muss es insgesamt abschließend geprüft werden.

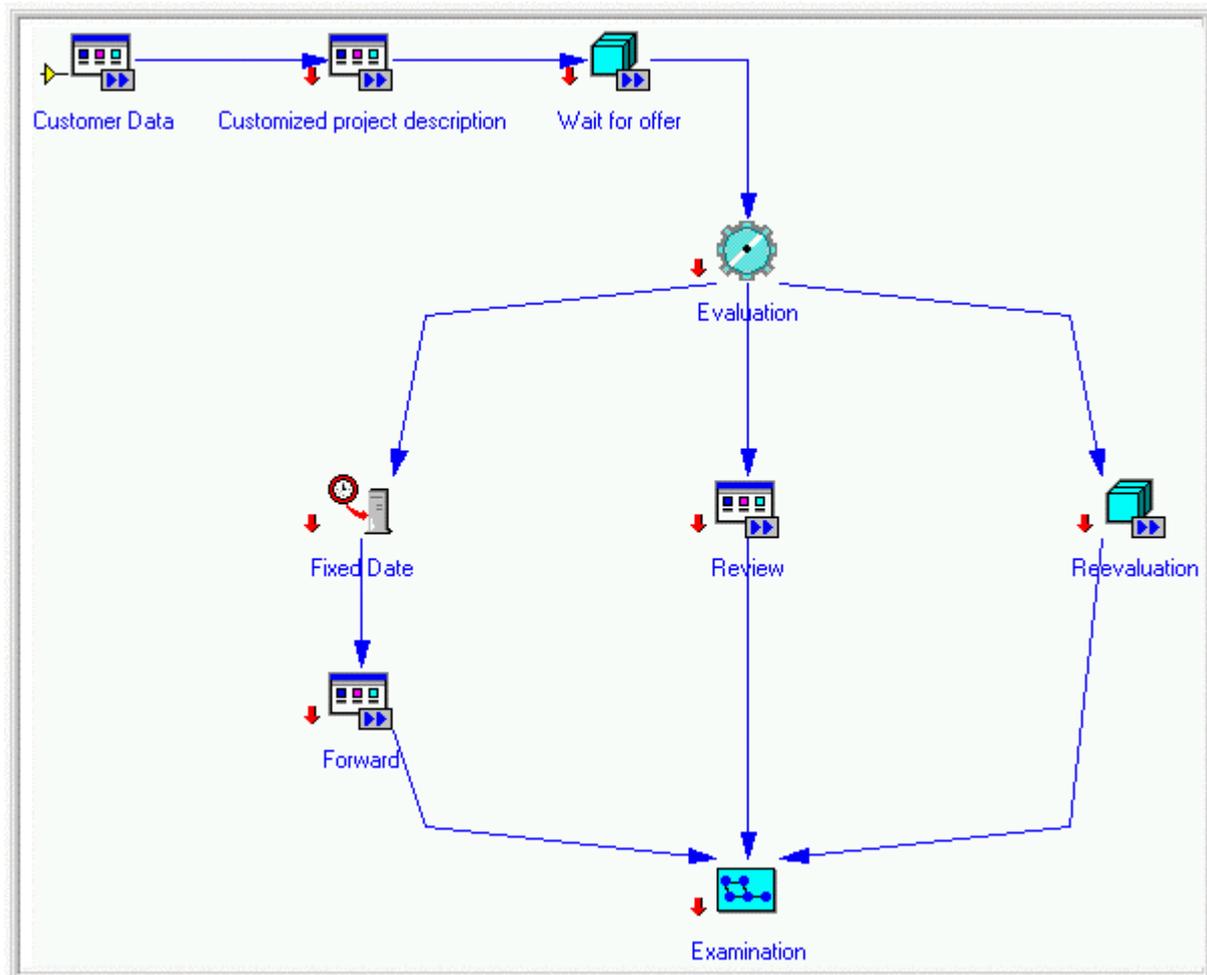


Abbildung 1: Beispiel für einen Prozess mit File und Time Events

2 Installation

Dieses Kapitel erörtert die Voraussetzungen, die Installation und die Konfiguration des Workflow Event Servers auf Microsoft Windows 2000 basierenden Systemen.

2.1 Voraussetzungen

Der Workflow Event Server ist eine Erweiterung des IBM Produktes MQSeries Workflow, deshalb ist eine installierte und konfigurierte Version von MQSeries Workflow Grundvoraussetzung.

Weiterhin wird für den Betrieb des Workflow Event Servers ein Java Runtime Environment benötigt, das kompatibel zu Sun J2SE 1.4. ist. Der Java Interpreter muss in der Systemvariable %PATH% angegeben sein. Dies geschieht bei einer Standardinstallation von Sun J2SE oder J2EE automatisch. Falls dies nicht der Fall ist, ist es notwendig, dass der Pfad des Java Interpreters bei jedem Start mit angegeben wird.

Weitere Voraussetzungen werden in *IBM MQSeries Workflow Installation Guide* ausführlich erläutert.

2.2 Installation

Das Installationspaket enthält drei *.jar Pakete. `mqwfes.jar` ist das Programmpaket (Event Server Time und File), die beiden anderen sind die Klassenpakete für die MQSeries Workflow Java API, die auch im Lieferumfang von MQSeries Workflow enthalten sind. Diese beiden Pakete müssen sich im Klassenpfad des verwendeten JRE befinden, also entweder

- in einem Verzeichnis liegen, das in den Systemvariablen %CLASSPATH% enthalten ist.
- in einem Verzeichnis liegen, das beim Aufruf von Workflow Event Server durch den Parameter `-cp` von `java.exe` angegeben wird (siehe *Starten des Workflow Event Server* Seite 6).
- im Standardklassenpfad des JRE liegen (`/$JAVA_HOME/jre/lib/ext`).

Die letzte Möglichkeit (Standardklassenpfad SRE) wird empfohlen.

Das Programmpaket `mqwfes.jar` muss lokal auf dem Server in ein Verzeichnis abgelegt werden. Es ist wichtig, dass der Pfad beim Starten des Servers bekannt ist und dieser bei dem Aufruf angegeben wird (siehe *Starten des Workflow Event Server* Seite 6). Damit ist die Installation abgeschlossen. Im weiteren wird erklärt, wie der Workflow Event Server konfiguriert und gestartet wird.

2.3 Starten des Workflow Event Server

Der folgende Abschnitt beschreibt die zwei Varianten den Workflow Event Server zu starten.

Die erste Variante erfolgt mittels Direktaufruf. Hierbei werden die notwendigen Parameter beim Start dem Server direkt übergeben.

Die Zweite geschieht mit Hilfe einer Konfigurationsdatei, in der die Informationen gespeichert sind. Bei dieser Variante werden die Daten bei dem ersten Start abgefragt und abgespeichert, und bei den dann folgenden Starts direkt ausgelesen.

Der Standardaufruf des Workflow Event Servers (der Java Interpreter ist standardmäßig installiert und das Installationspaket wie empfohlen installiert):

```
java -jar mqwfes.jar
```

Befinden sich die MQSeries Workflow Java API Klassenpakete nicht im Standardklassenpfad des JRE so lautet der Aufruf:

```
java -jar -cp <Pfad der Klassenpakete> mqwfes.jar
```

Ein Aufruf mit den Grundparametern sieht wie folgt aus:

```
java -jar mqwfes.jar -u <User> -s <Service>1
```

2.4 Konfiguration

Der Workflow Event Server bietet zwei Konfigurationsmöglichkeiten.

- Parameterangaben beim Start

(Parameter siehe *Tabelle 1: Systemparameter* Seite 8 und Syntax siehe *Starten des Workflow Event Server* Seite 6). Die Parameter haben Vorrang vor den in der Konfigurationsdatei angegebenen Werten. Die bereits vorhandenen Parameter werden nicht überschrieben.

- Starten mit Konfigurationsdatei *mqwfes.ini*.

Die Konfigurationsdatei kann während dem Starten angelegt werden, falls die Parameter nicht im Aufruf enthalten sind. In der Konfigurationsdatei sind die für den Betrieb wichtigen Systemparameter eingetragen.

Eine Zeile ist dabei folgendermaßen aufgebaut:

```
Parameter = Wert  
(KEINE Zeilenbegrenzer, z.B. ';' )
```

Eine Zeile kann nur ein Schlüssel / Wert-Paar aufnehmen, weitere werden ignoriert. Kommentare können mit '#' angefügt werden.

¹ Siehe Kapitel Systemparameter

Beim Programmstart ohne Parameter wird das Vorhandensein eines Konfigurationsfiles geprüft. Sofern nicht in der Kommandozeile anders angegeben, wird in dem Verzeichnis nach `mqwfes.ini` gesucht, in dem sich das Programmpaket befindet.

Wird die Datei nicht gefunden, bietet das Programm die Möglichkeit, die Datei zu erstellen, wobei dann die gewünschten Werte abgefragt werden. Wird diese Möglichkeit nicht gewählt, werden die Parameter der Kommandozeile ausgewertet. Sind diese für den Betrieb nicht ausreichend, wird das Programm mit entsprechender Fehlermeldung beendet.

2.4.1 Systemparameter

Die folgende Tabelle zeigt eine Übersicht von Programmparametern, ihren Wertebereichen, Defaultwerten, den Schlüsselworten in der Konfigurationsdatei, Kommandozeilenparameter und ob sie mit dem `set` Kommando (Erklärung des Kommandos: `set` siehe Seite 24) verändert werden können.

Bezeichnung im Inifile	Wertebereich	Defaultwert	Parameter	Set² Kommando	Erläuterung
login	String	kein	-u	-	Benutzername
password	String case-sensitive	kein	-p	-	Passwort im Klartext
service	String „DELAY“, „DIRECTORY“	kein	-s	-	Es muss angegeben werden, welches Event überwacht werden soll
-	boolean	true	-	Logging	Gibt an, ob in das Logfile geschrieben werden soll. Standardmäßig werden Nachrichten mitgeschrieben.
logsize	integer	100	-	Logsize	Logsize legt die maximal Größe des Logfiles fest. Sobald diese überschritten wird, werden alle bis auf die letzten 100 Einträge gelöscht. Die Größe wird in Kb angegeben. Empfohlen wird logsize auf mindestens 6Kb zu setzen, da 100 Zeilen ca. 5Kb groß sind.
Log_commands	boolean	false	-	log_commands	Schreibt die an der Konsole eingegebenen Kommandos in das Logfile.
log_messages	boolean	false	-	log_messages	Schreibt die Informationsausgaben des Programms in die Logfile
priority1	integer	60	-p1	priority1	Priority1 ist die höchste Priorität. Sie wird in Sekunden angegeben.
priority2	integer	300	-p2	priority2	Priority2 ist die zweit höchste Priorität. Sie

² Siehe Kapitel 4.2.9 „set“

Bezeichnung im Inifile	Wertebereich	Defaultwert	Parameter	Set² Kommando	Erläuterung
					wird in Sekunden angegeben.
priority3	integer	900	-p3	priority3	Priority3 wird in Sekunden angegeben.
priority4	integer	1800	-p4	priority4	Priority4 wird in Sekunden angegeben.
priority5	integer	3600	-p5	priority5	Priority5 ist die zweit niedrigste Priorität. Sie wird in Sekunden angegeben.
priority6	integer	86400	-p6	priority6	Priority6 ist die niedrigste Priorität. Sie wird in Sekunden angegeben.
refresh	integer	kein	-r	refresh	Refresh gibt das Intervall an, in dem die Worklisten aktualisiert werden. Es wird in Minuten angegeben.
logfile	Pfad	„mqwfes.log“	-	logfile	Logfile gibt den vollständigen Pfad und den Namen des Logfiles an.
-	Pfad	„mqwfes.ini“	-i	-	Inifile gibt den vollständigen Pfad und den Namen des Inifile an.
create_directories	boolean	false	-	create_directory	Legt für das File Event Zielverzeichnisse für die Funktionen „Kopieren“ und „Verschieben“ automatisch an, falls diese nicht vorhanden sind.
threshold	integer	300	-	threshold	Threshold gibt an, wie viele Events in die Workliste aufgenommen werden können.
tracing	boolean	false	-	tracing	Tracing schreibt alle Informationen und Fehler in Logfile zusammen mit einer genauen Zeitangabe

Tabelle 1: Systemparameter

Einige der Systemparameter (siehe *Tabelle 1: Systemparameter*) können im laufenden Betrieb mittels des `set` Kommandos verändert werden, wobei die Werte im Konfigurationsfile erhalten bleiben.

3 Modellieren in MQSeries Workflow

Nachdem das vorhergehende Kapitel sich mit der Installation des Workflow Event Servers beschäftigt hat, wird in diesem das Einbinden von File und Time Events in MQSeries Workflow durch das Programm Buildtime erläutert.

3.1 Workflow Konfiguration

Um den Workflow Event Server in einem Workflow System verwenden zu können, müssen folgende Einstellungen in MQSeries Buildtime getätigt werden.

- Anlegen der User: Für jeden Eventserver ist ein User anzulegen
- Erteilen der Rechte „operation administration“ für die betreffende Kategorie und „Persons: Workitem“ ? „all users“
- Optional: Einbinden der Icons für die Events File und Time

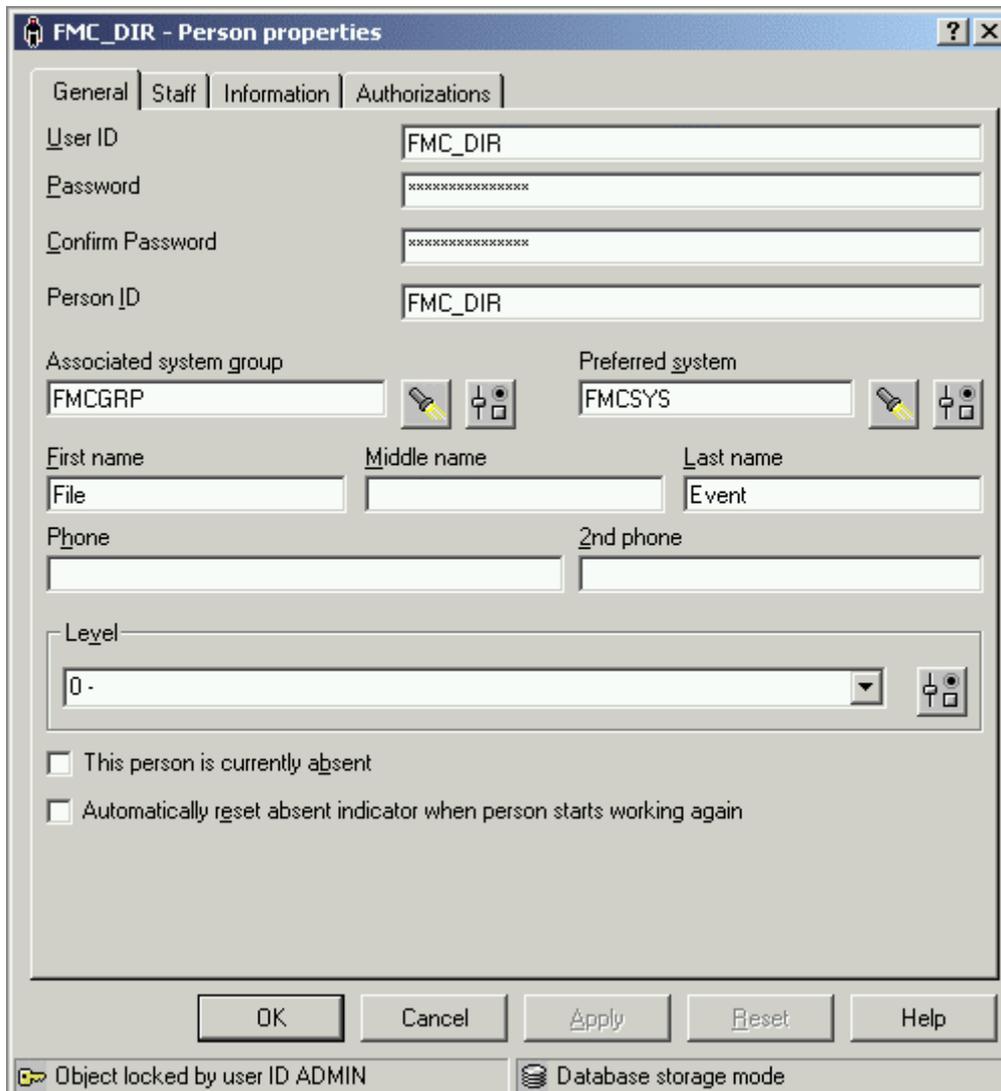
3.1.1 Beispiel für das Event File

Der folgende Abschnitt behandelt die notwendigen Einstellungen anhand des File Events, die Einstellungen für das Time Event sind identisch. Im Beispiel heißt der Time Event „FMC_DELAY“ und der File Event „FMC_DIR“.



Der erste Schritt ist das Anlegen eines neuen Users in Buildtime unter „Staff“ mittels Rechtsklick auf „Persons“, „new Person“.

Abbildung 2: Anlegen eines Users



FMC_DIR - Person properties

General | Staff | Information | Authorizations

User ID: FMC_DIR

Password: *****

Confirm Password: *****

Person ID: FMC_DIR

Associated system group: FMCGRP

Preferred system: FMCSYS

First name: File

Middle name:

Last name: Event

Phone:

2nd phone:

Level: 0-

This person is currently absent

Automatically reset absent indicator when person starts working again

OK Cancel Apply Reset Help

Object locked by user ID ADMIN Database storage mode

Abbildung 3: Eintragen der Benutzerinformationen

Unter den „Properties“ werden die Benutzerinformationen eingetragen, wobei zu beachten ist, dass die User ID und das Passwort mit dem in der `mqwfes.ini` oder mit den angegebenen Parametern übereinstimmen.

3.1.2 Rechtemanagement

Der Benutzer benötigt besondere Rechte um auf den Server zugreifen zu können.

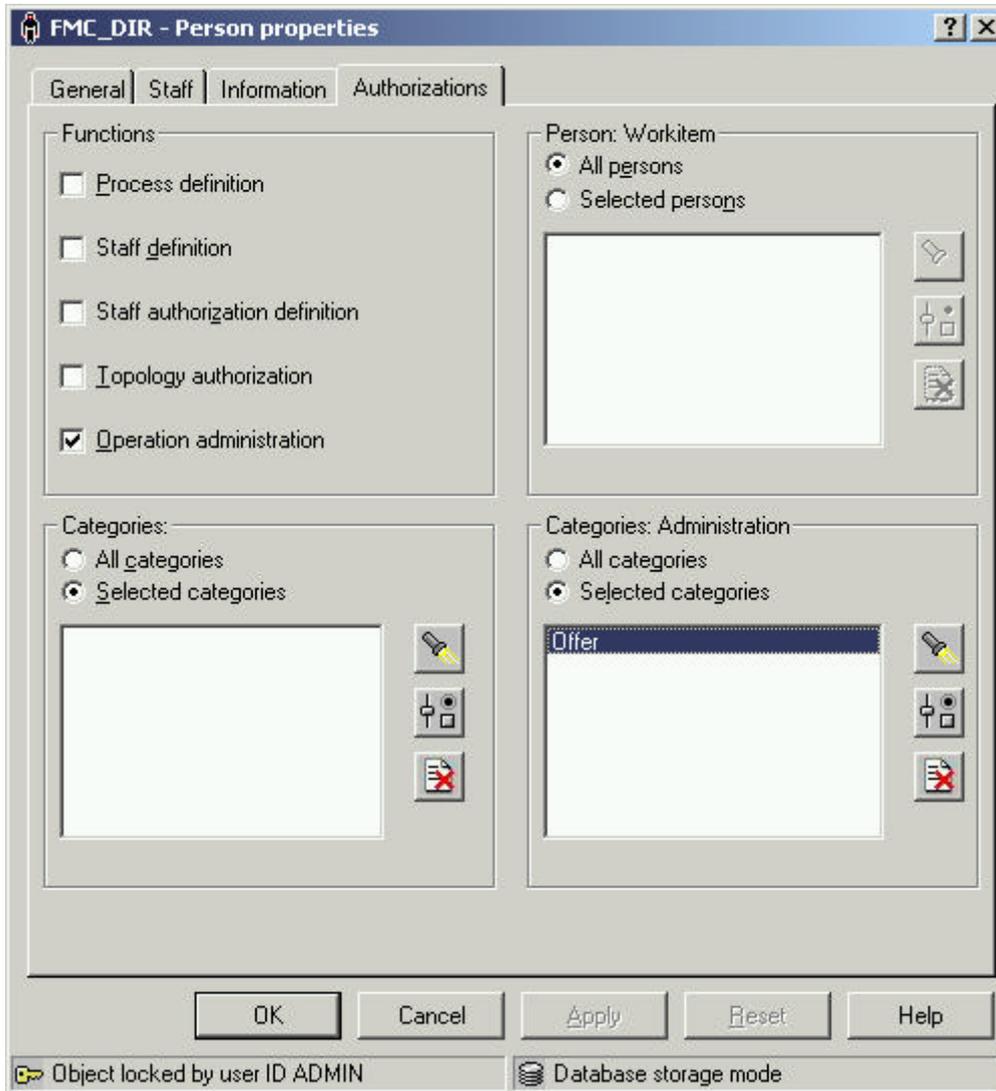


Abbildung 4: Zuweisen von Rechten

In den Eigenschaften des Benutzers wird unter dem Punkt „Authorizations“ das „Operation administration“ Recht zugewiesen.

Des weiteren muss er unter „Workitem“ die Rechte für „All persons“ bekommen und unter „Administration“ den entsprechenden Kategorien zugewiesen werden.

3.2 Workflow Design

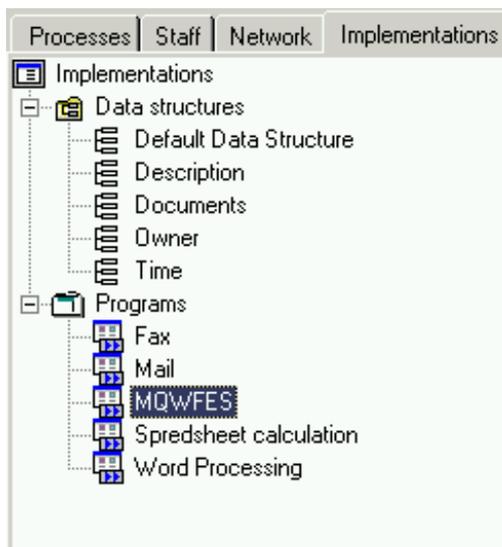
Um den Workflow Event Server in einen Workflow Prozess einzubinden, muss an der entsprechenden Stelle des Prozesses im Buildtime eine Aktivität erzeugt werden. Das Erzeugen von Aktivitäten wird ausführlich in *MQ Workflow Concepts and Architecture* und *Getting Started with Buildtime* erklärt.

Die Event Aktivitäten benötigen folgende Eigenschaften:

- Start manual
- Exit automatic
- Unter Staff muss explizit der Benutzer des Workflow Event Server angegeben sein (oder im Fall der späteren Zuweisung Prozessadministrator)
- Im Beschreibungsfeld muss **DELAY** oder **DIRECTORY** als Schlüsselwort mit den Parameter angegeben werden
- Ein implementiertes Aktionsprogramm

3.2.1 Anlegen des Programms

Der erste Schritt im Design mit Buildtime ist das Anlegen eines Programms. Dieses wird später unter die Events gelegt. Da das Programm in der aktuellen Implementierung nicht gestartet wird, ist es unerheblich, welches Programm hierzu genutzt wird.



Als erstes wird in Implementations unter Programs ein neues Programm, hier MQWFES genannt, angelegt.

Abbildung 5: Anlegen eines Programms

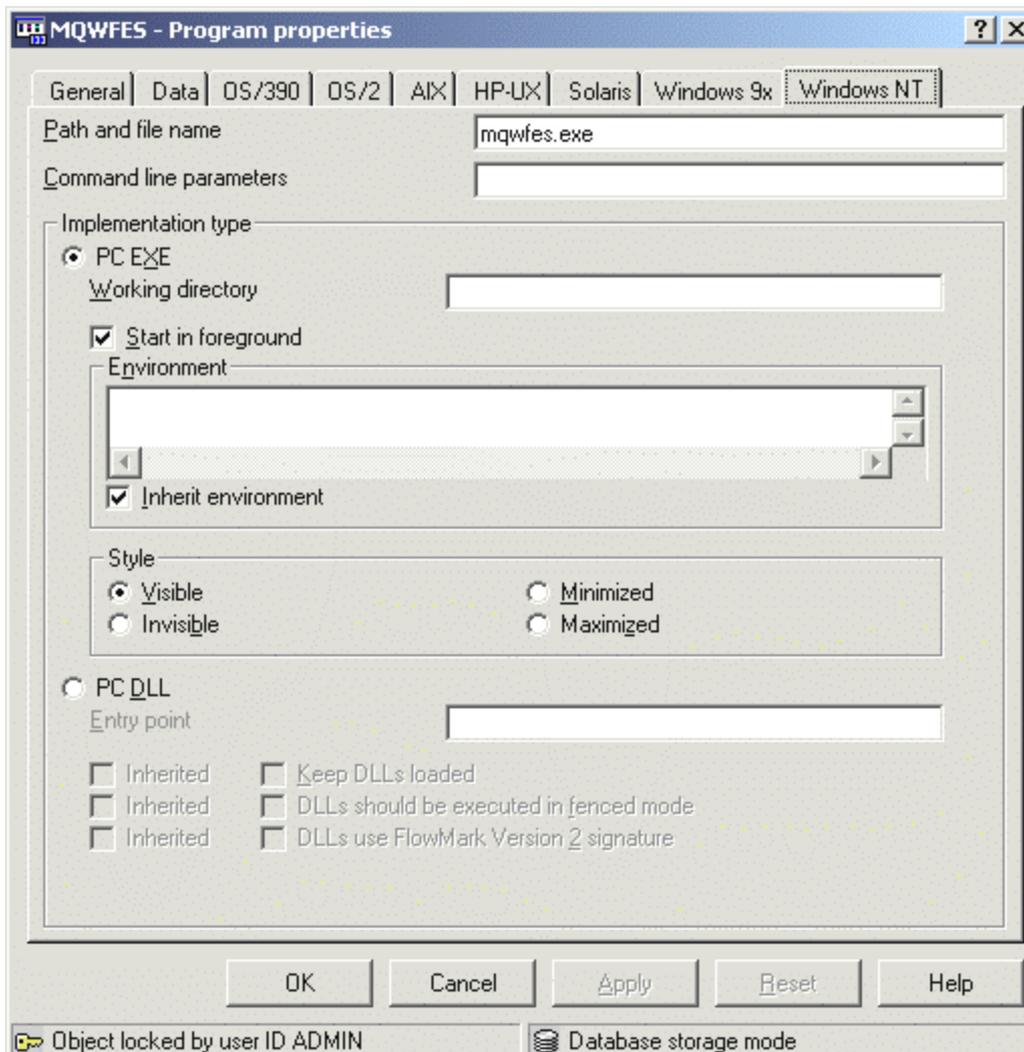


Abbildung 6: Eintragen der Eigenschaften

In diesem Beispiel wird gezeigt dass nur ein syntaktisch korrektes (aber nicht unbedingt ein real existierendes) Programm in den Eigenschaften eingetragen werden muss.

Weiterhin ist zu beachten, dass die Datenstruktur dem Prozess angepasst werden muss. Die einfachste Methode um sicherzustellen, dass das Programm universal einsetzbar ist, ist unter dem Punkt Data „*Program can handle any data structures*“ auszuwählen. Anderenfalls muss die Datenstruktur mit der des übergeordneten Container übereinstimmen.

3.2.2 Einbinden von Icons

Für die Events File und Time verwendeten Icons sind und im Programmpaket enthalten. Um sie zu verwenden, müssen sie in die Buildtimedatenbank importiert werden. Das Importieren in die Buildtimedatenbank erfolgt im Buildtime.

Es müssen folgende Schritte durchgeführt werden:

1. Anlegen eines neuen Prozesses oder öffnen eines bestehenden Prozesses
2. Process Diagramm Fenster aktivieren
3. In der Menüleiste unter *View, Drawing Tools* den Punkt *Customize* auswählen (siehe *Abbildung 7: Auswählen von Drawing Tool Sets* Seite 14)
4. Mit *Add Icons Einfügen* (siehe *Abbildung 8: Drawing Tool Sets (1)* Seite 15)
5. Mit *Drag & Drop* auf „*Additional Icons for this Activity Type*“ in „*Program Activity*“ ziehen (siehe *Abbildung 9: Drawing Tool Sets (2)* Seite 15)
6. Mit *OK* bestätigen (siehe *Abbildung 10: Eingebundene Icons* Seite 16)

Um die im Beispiel verwendeten Icons zu verwenden, müssen sie unter den Namen MQWFESF und MQWFEST in die Datenbank importiert sein.

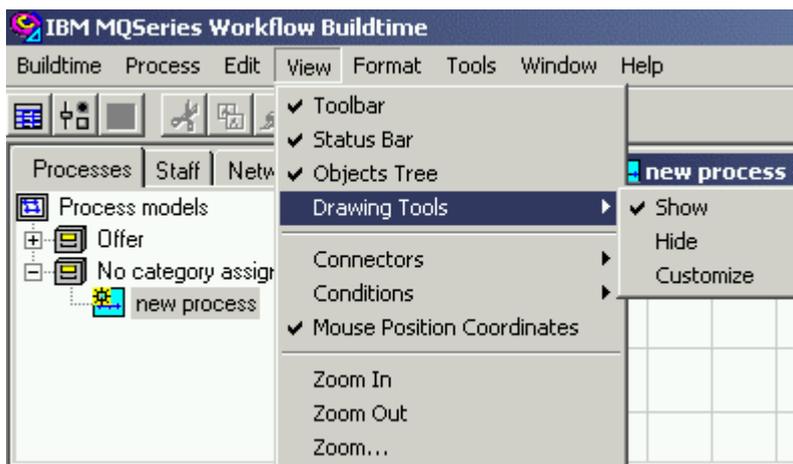


Abbildung 7: Auswählen von Drawing Tool Sets

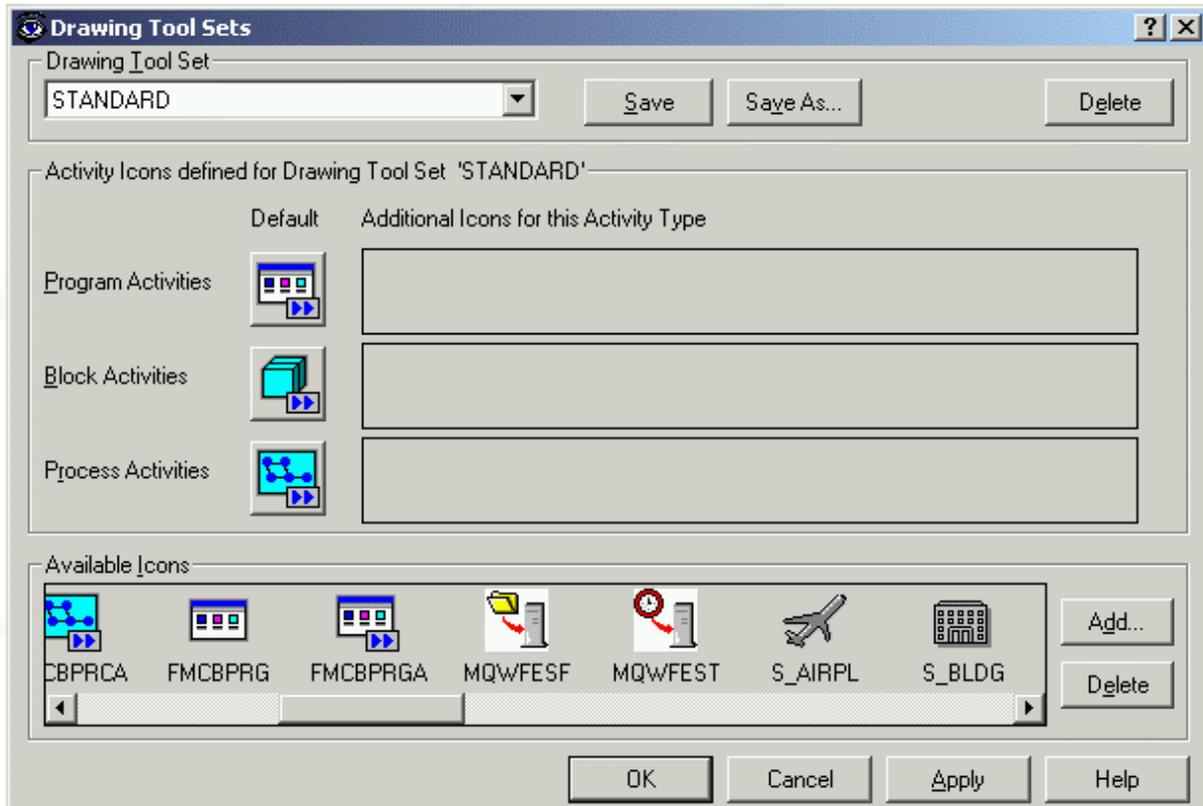


Abbildung 8: Drawing Tool Sets (1)

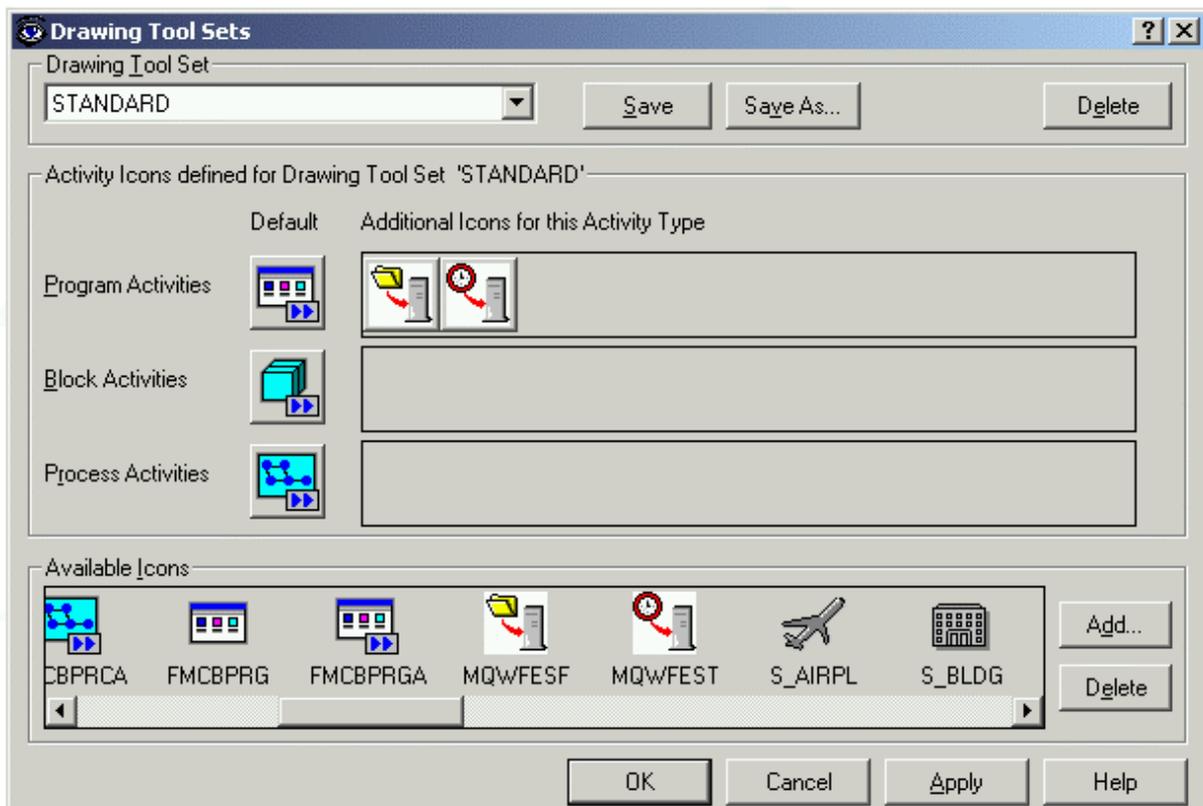


Abbildung 9: Drawing Tool Sets (2)

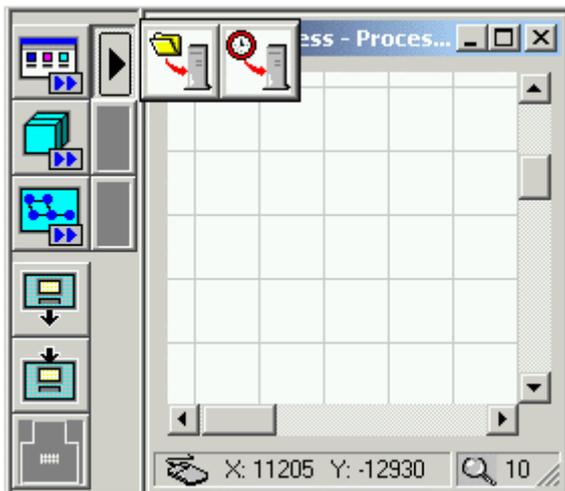


Abbildung 10: Eingebundene Icons

3.2.3 Anlegen einer Aktivität

In dem Prozess wird nun eine ganz normale Programmaktivität angelegt.

In den Eigenschaften dieser Aktivität werden nun unter „General“ die benötigten Parameter eingetragen. Es ist zu beachten, dass das in der *mqwfes.ini* angegebene Schlüsselwort am Anfang der Beschreibung steht (DIRECTORY oder DELAY). Sollte es nicht vorhanden sein, wird diese Aktivität vom Server nicht bearbeitet.

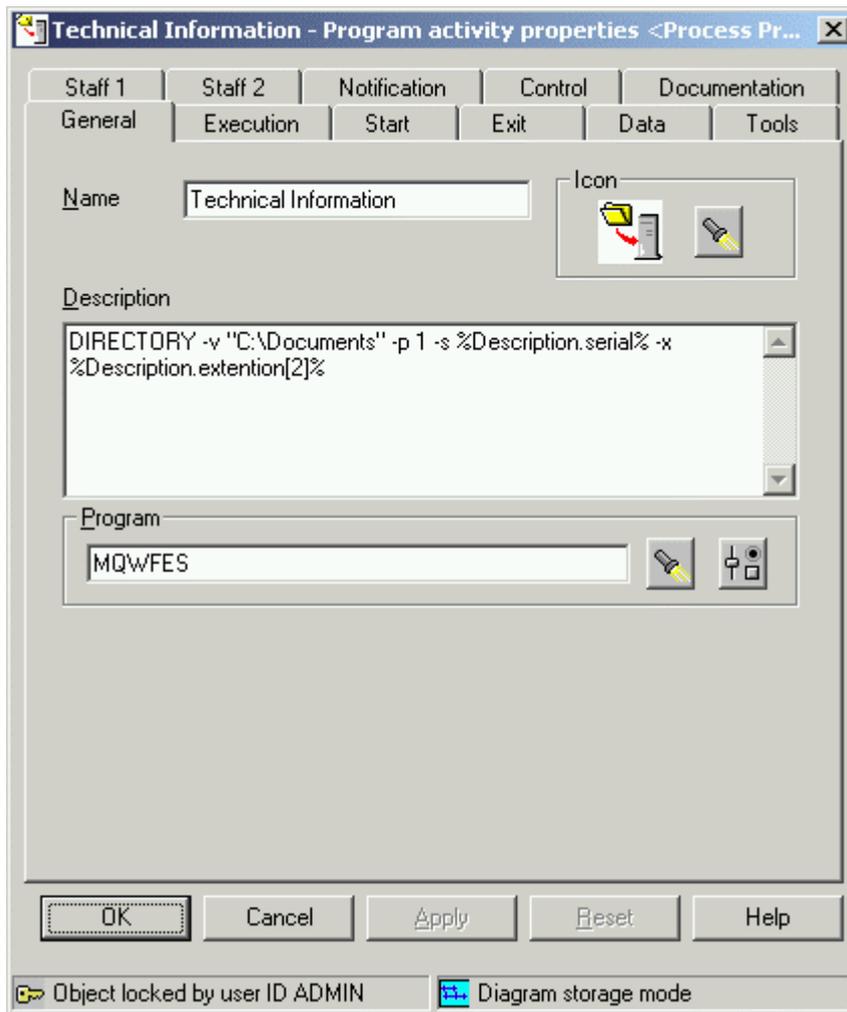
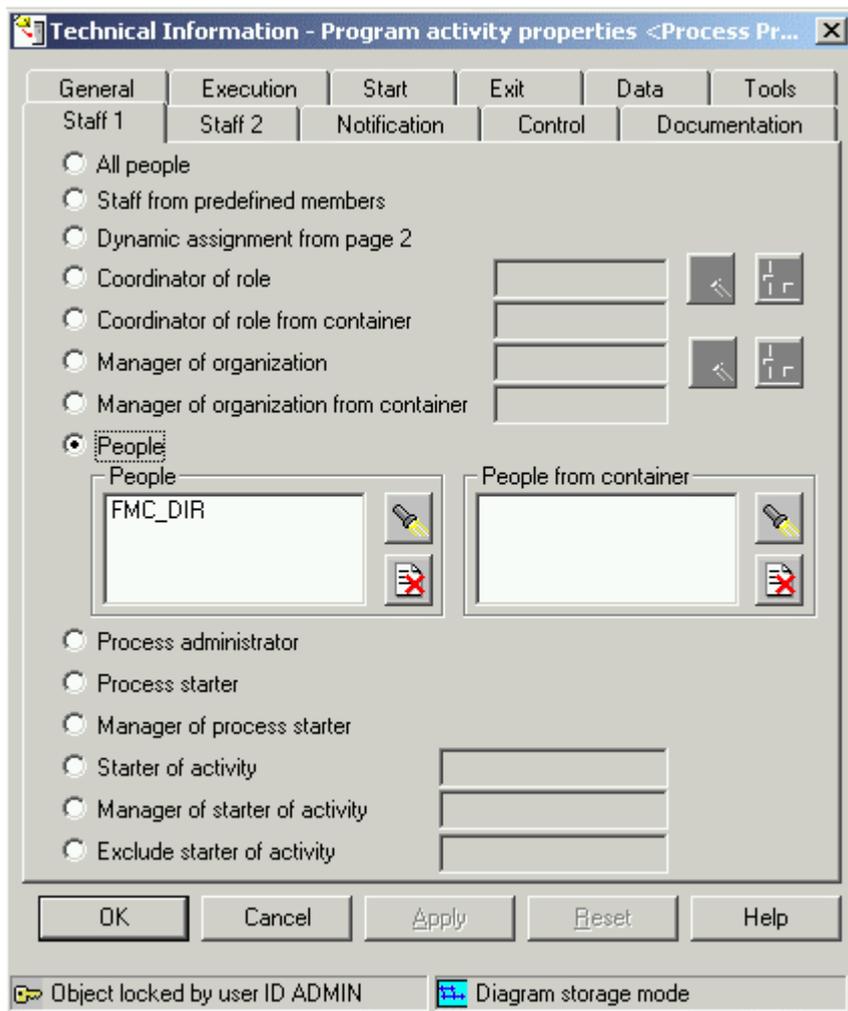


Abbildung 11: Attribute der Aktivität (1)

Parameterwerte können auch einer Datenstruktur entnommen werden. Im Beschreibungsfeld der *Abbildung 11: Attribute der Aktivität (1)* wird die Dateibeschreibung der Datenstruktur entnommen. Dies geschieht mittels Platzhaltern. Das gewünschte Feld der Datenstruktur wird angegeben und als Befehl mit ‚%‘ umschlossen. (Nähere Informationen zum Angeben der Parameter befinden sich unter den Punkt *Zugehöriger Wert 6.2 Parameterübersicht* siehe Seite 27)

Der Name der Aktivität ist frei wählbar.

Unter „*Program*“ muss das unter Abschnitt „*Anlegen des Programm*“ Seite 12 erstellte Programm ausgewählt sein.

**Abbildung 12: Attribute der Aktivität (2)**

In „Staff 1“ unter „People“ darf nur der angelegte Benutzer ausgewählt werden (siehe *Abbildung 12: Attribute der Aktivität (2)* Seite 18).

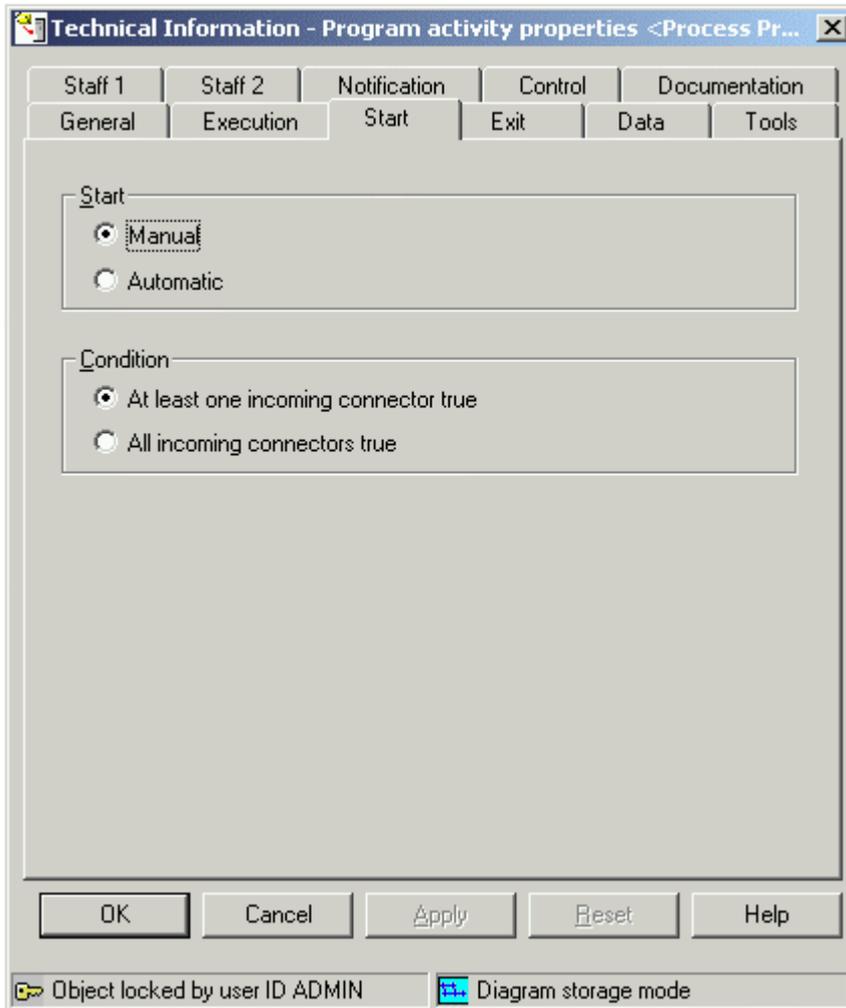


Abbildung 13: Attribute der Aktivität (3)

In der letzten Einstellung ist es notwendig die Aktivität auf „Manual“ einzustellen.

3.2.4 Durchreichen von Datenstrukturen

Mit der Hilfe von Datenstrukturen können Informationen übermittelt werden (siehe Kapitel 8.2.2 „Durchführung“ Seite 33). Nur wenn eine Aktivität gestartet wurde, kann aktiv in die Datenstruktur geschrieben werden.

Der Workflow Event Server startet im Rahmen der Eventbehandlung keine Aktivitäten. Zur Weiterleitung von Daten, dem Lesen aus dem Inputcontainer und das Schreiben in den Outputcontainer, wird der Data Default Connector benutzt.

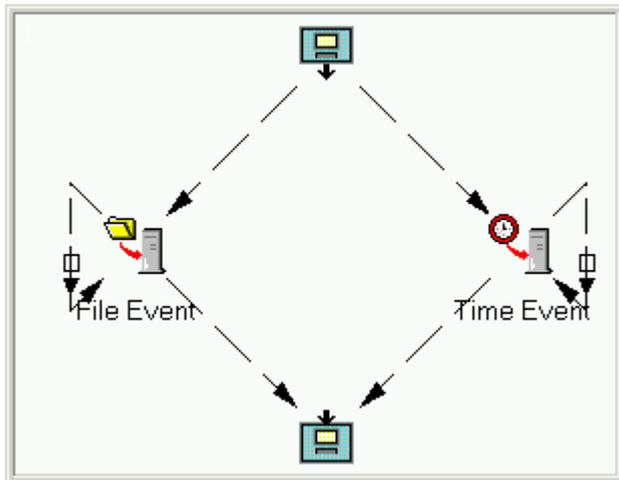


Abbildung 14: Datenkonnektoren

4 Kontrollmöglichkeiten durch den Administrator

Der Administrator hat zwei Möglichkeiten in den Prozessablauf einzugreifen. Als Prozessadministrator hat er die Möglichkeit, Items auf die Worklisten von der Event Server zu setzen. In der Konsole kann er sowohl die Eigenschaften als auch die Items ändern. Durch diese Möglichkeiten können fehlerhafte Items korrigiert und der reibungslose Prozessablauf sichergestellt werden. Dieses Kapitel beschäftigt sich genauer mit diesen zwei Möglichkeiten.

4.1 Administration im Client

Der im Buildtime erstellte Prozess wird am Client gestartet. Der hier verwendete Client ist der mitgelieferte Standardclient. Häufig werden für firmeninterne Anwendungen eigens erstellte Masken verwendet, aber die grundlegende Funktionalität ändert sich nicht. In diesem Abschnitt wird die Kontrollmöglichkeiten im Client dargestellt (weitere Informationen sind in *Getting started with Runtime* nach zu lesen).

4.1.1 Importieren in Runtime

Bevor der im Buildtime erstellte Prozess am Client gestartet werden kann, muss dieser in die Runtimekomponente importiert werden. Das Importieren erfolgt durch folgenden Konsolenaufwurf:

```
C:\>fmcibie -ppassword -uADMIN -ot -i"<Pfad>\<Name>.fdl"
```

Jetzt kann der Client gestartet werden. Detaillierte Informationen zum Import sind im Handbuch *MQSeries Workflow: Getting started with Runtime* nachzulesen.

Benutzerdefinierte Icons müssen vor dem Import im System bekannt sein.

4.1.2 Arbeiten im Client

Um den Benutzern zu ermöglichen Prozesse zu starten, muss der Prozessadministrator einige Konfigurationsmaßnahmen durchführen. Standardmäßig ist der Prozessadministrator als Benutzer „Admin“ mit „password“ als Passwort definiert. Im geöffneten Fenster *Baumstruktursicht* wird der Prozess auf drei unterschiedlichen Ebenen dargestellt.

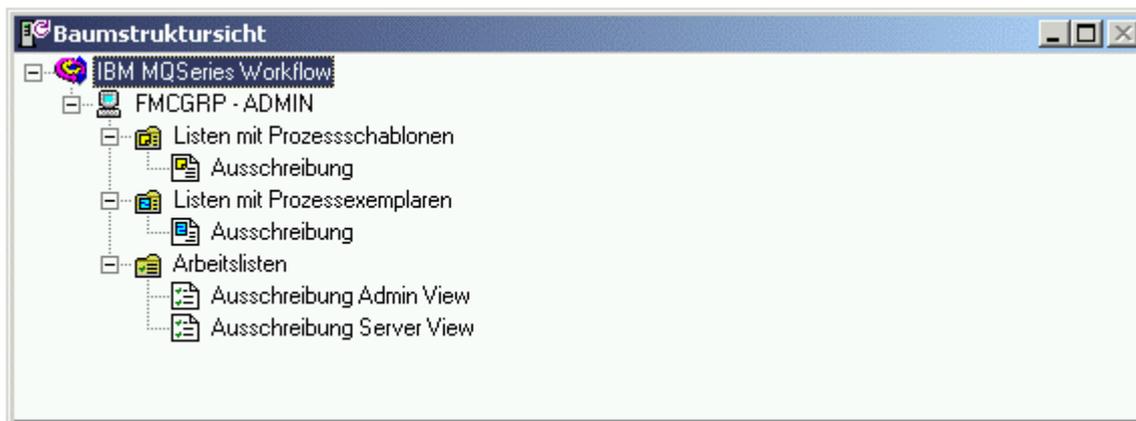


Abbildung 15: Baumstruktur im Client

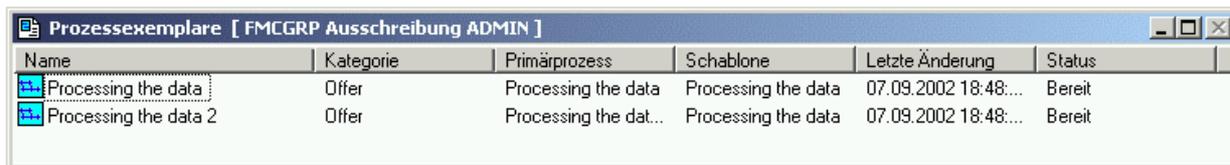
Im ersten Schritt muss der Prozessadministrator eine Liste mit Prozessschablonen, eine Liste mit Prozessexemplaren und eine Arbeitsliste im Fenster *Baumstruktur* erstellen.



Name	Beschreibung	Kategorie	Erstellungszeit	Letzte Änderung	Gültig ab
Examination		Offer	07.09.2002 18:44:18	07.09.2002 18:44:18	07.09.2002 18:44:18
Processing the data		Offer	07.09.2002 18:44:20	07.09.2002 18:44:20	07.09.2002 18:44:19

Abbildung 16: Prozessschablone mit Prozessen

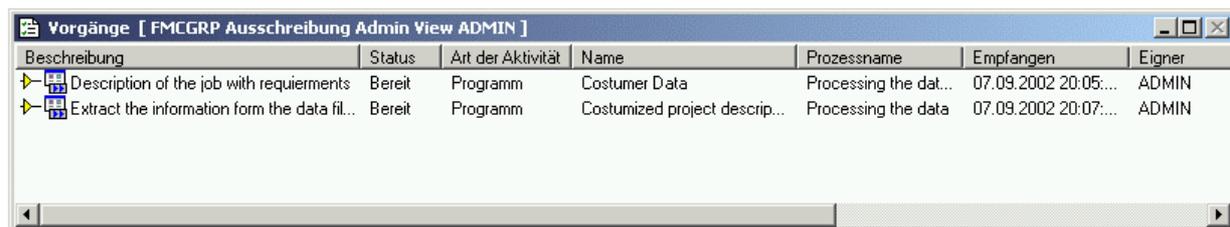
Aus Prozessschablonen können Prozessexemplare mehrfach erzeugt werden. Alle laufenden Prozesse können in der Liste Prozessexemplare gefunden werden.



Name	Kategorie	Primärprozess	Schablone	Letzte Änderung	Status
Processing the data	Offer	Processing the data	Processing the data	07.09.2002 18:48:...	Bereit
Processing the data 2	Offer	Processing the dat...	Processing the data	07.09.2002 18:48:...	Bereit

Abbildung 17: Prozessexemplar Processing the data

Im Arbeitslisten (Vorgänge) werden die aktuellen Aktivitäten dargestellt. Von diesem Punkt aus kann ihr Status überwacht werden, aber auch verändert werden. Ein Aktivität kann gestartet, neugestartet, aber auch beendet werden.



Beschreibung	Status	Art der Aktivität	Name	Prozessname	Empfangen	Eigner
Description of the job with requierments	Bereit	Programm	Costumer Data	Processing the dat...	07.09.2002 20:05:...	ADMIN
Extract the information form the data fil...	Bereit	Programm	Costumized project descrip...	Processing the data	07.09.2002 20:07:...	ADMIN

Abbildung 18: Arbeitsliste mit Aktivitäten

4.1.3 Fehlerbehandlung

In diesem Abschnitt werden die Möglichkeiten beschrieben, die der Prozessadministrator hat, mit fehlerhaften Aktivitäten zu verfahren.

4.1.3.1 Übertragen des Prozesses durch den Administrator

Ist ein Time oder File Event syntaktisch falsch, wird dieses als fehlerhaft auf die Workliste (Arbeitsliste) des Prozessadministrators gesetzt. Jetzt hat der Prozessadministrator die Möglichkeit, das Beschreibungsfeld des Items zu ändern und somit zu berichtigen. Um das Beschreibungsfeld zu ändern müssen die Eigenschaften (über Kontextmenü der rechten Maustaste) der entsprechenden Aktivität geöffnet werden. Ist die Änderung erfolgt, kann das Item auf die Workliste der Event Server verschoben werden. Ist ein Item zwar syntaktisch richtig aber inhaltlich falsch, müssen diese Änderungen von der Konsole aus vorgenommen werden (siehe 4.2.2 *change* Seite 23).

4.1.3.2 Abbrechen Erzwingen

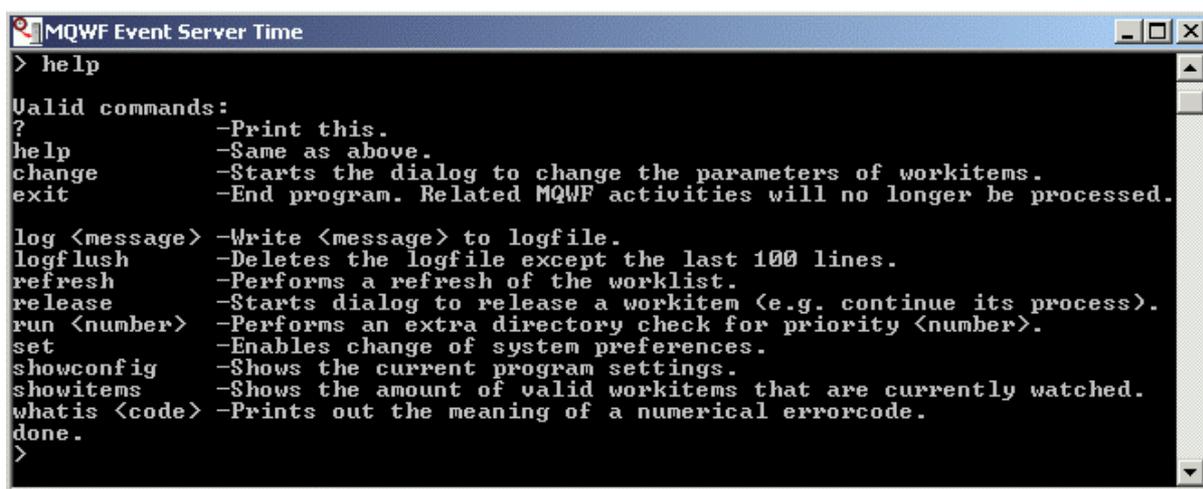
Befindet sich auf der Workliste des Prozessadministrators eine Aktivität, die nicht normal beendet werden kann, besteht auch die Möglichkeit mit der Option „Abbrechen Erzwingen“ diese zu beenden.

4.2 Kommandos für die Konsole

Der Event Server wird in einem Eingabeprompt gestartet. Auf der Konsole werden standardmäßig die wichtigsten Statusmeldungen dargestellt, Details werden direkt in `mqwfes.log` geschrieben. Alle Informationen können aber auch mit Kommandos auf der Konsole ausgegeben werden. Dieser Abschnitt erläutert die zur Verfügung stehenden Befehle.

4.2.1 „?“ oder help

Mit `?` oder `help` kann eine Auflistung aller Befehle angefordert werden. Sämtliche Befehle werden mit einer Kurzbeschreibung auf der Konsole ausgegeben. Damit wird das Navigieren vereinfacht.



```
MQWF Event Server Time
> help
Valid commands:
?          -Print this.
help       -Same as above.
change     -Starts the dialog to change the parameters of workitems.
exit       -End program. Related MQWF activities will no longer be processed.

log <message> -Write <message> to logfile.
logflush   -Deletes the logfile except the last 100 lines.
refresh    -Performs a refresh of the worklist.
release    -Starts dialog to release a workitem (e.g. continue its process).
run <number> -Performs an extra directory check for priority <number>.
set        -Enables change of system preferences.
showconfig -Shows the current program settings.
showitems  -Shows the amount of valid workitems that are currently watched.
whatis <code> -Prints out the meaning of a numerical errorcode.
done.
>
```

Abbildung 19: show help im Eingabeprompt

4.2.2 change

Startet einen Dialog, um die Parameter eines Workitems zu ändern. Abgefragt werden die Kategorie, Prozessinstanz, Name der Aktivität und die neuen Parameter, wobei hier alle zur Verfügung stehenden Optionen aufgeführt werden.

4.2.3 exit

`Exit` dient zum geregelten Shutdown des Servers. Damit wird die Abarbeitung der Events gestoppt.

4.2.4 log <message>

Dieses Kommando dient zum Schreiben von Nachrichten in die Logdatei. Dies kann hilfreich sein um besondere Ausgaben in der Logdatei wiederzufinden.

Die Syntax lautet: `log <message>`.

Wobei `<message>` der gewünschte Text ist, der mit einem „Enter“ abgeschlossen wird.

4.2.5 logflush

Die Logdatei hat eine festgelegte Größe. Der Standardwert ist hier 100KB. Dieser Wert kann aber auch vom Administrator in der Konfigurationsdatei beliebig festgelegt werden. Sobald die festgelegte

Dateigröße überschritten wurde, wird alles bis auf die letzten 100 Zeilen gelöscht. Diese Methode kann auch manuell im Eingabeprompt mit dem Befehl `<logflush>` angestoßen werden.

4.2.6 refresh

Mit `refresh` kann die aktuelle Workliste unabhängig von der Refreshzeit vom MQSeries Workflow Runtime Server geholt werden.

4.2.7 release

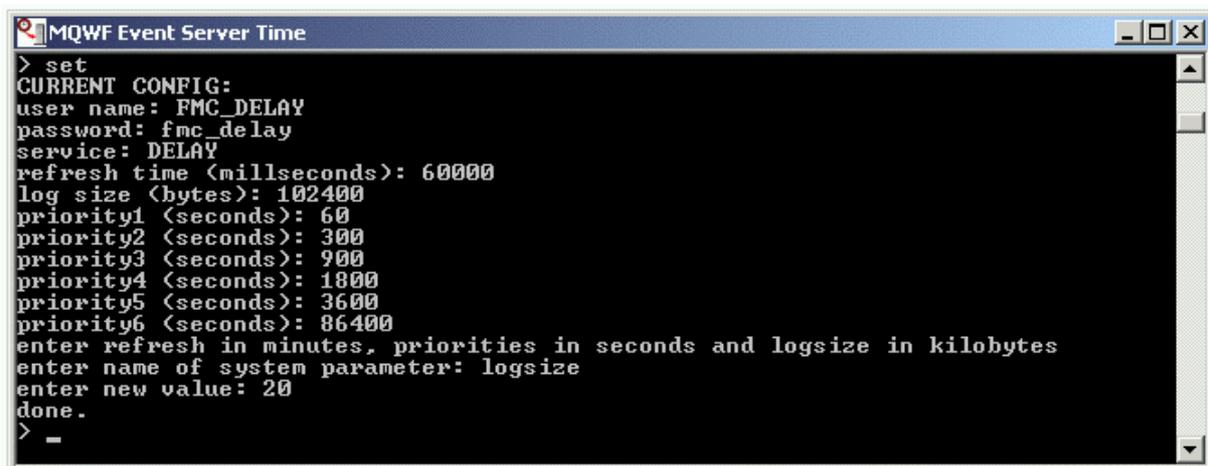
`Release` startet den Dialog um ein einzelnes Workitem manuell abzuarbeiten. Der zugehörige Prozess wird dann weitergeführt und das Workitem taucht nicht mehr in der Workliste auf.

4.2.8 run <number>

`Run` initialisiert die manuelle Überprüfung der Directory Items der gewünschten Priorität. Bei dem Wert 0 für `<number>` werden alle Prioritäten überprüft.

4.2.9 set

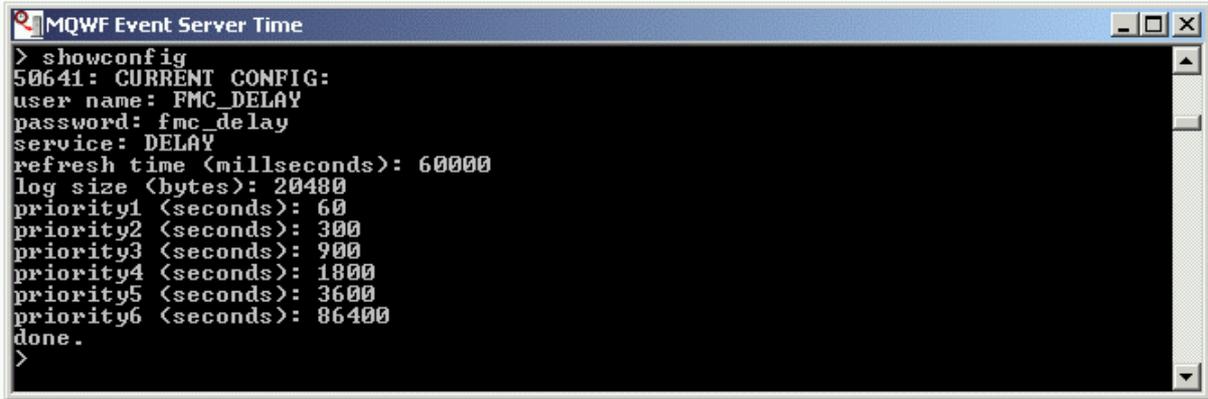
Das Kommando `set` bietet die Möglichkeit, die Programmparameter („Tabelle 1: Systemparameter“ siehe Seite 8) neu zu setzen. Diese Änderungen werden nicht in die Konfigurationsdatei übernommen. Die Änderung kann per Dialog geschehen, der mit `set` aufgerufen wird. Oder es kann mit `set <parameter>` ein Wert angezeigt werden und mit `set <parameter> <wert>` ein Wert geändert wird. Bei Eingabe von `set` werden alle Systemparameter aufgelistet.



```
MQWF Event Server Time
> set
CURRENT CONFIG:
user name: FMC_DELAY
password: fmc_delay
service: DELAY
refresh time (milliseconds): 60000
log size (bytes): 102400
priority1 (seconds): 60
priority2 (seconds): 300
priority3 (seconds): 900
priority4 (seconds): 1800
priority5 (seconds): 3600
priority6 (seconds): 86400
enter refresh in minutes, priorities in seconds and logsize in kilobytes
enter name of system parameter: logsize
enter new value: 20
done.
> -
```

Abbildung 20: set Dialog im Eingabeprompt

4.2.10 showconfig



```
MQWF Event Server Time
> showconfig
50641: CURRENT CONFIG:
user name: FMC_DELAY
password: fmc_delay
service: DELAY
refresh time (milliseconds): 60000
log size (bytes): 20480
priority1 (seconds): 60
priority2 (seconds): 300
priority3 (seconds): 900
priority4 (seconds): 1800
priority5 (seconds): 3600
priority6 (seconds): 86400
done.
>
```

Showconfig zeigt aktuellen Einstellungen des Programms an. Von den gezeigten Werten können refresh, logsize und die Prioritäten neu gesetzt werden (siehe set).

Abbildung 21: showconfig in Eingabeprompt

4.2.11 showitems

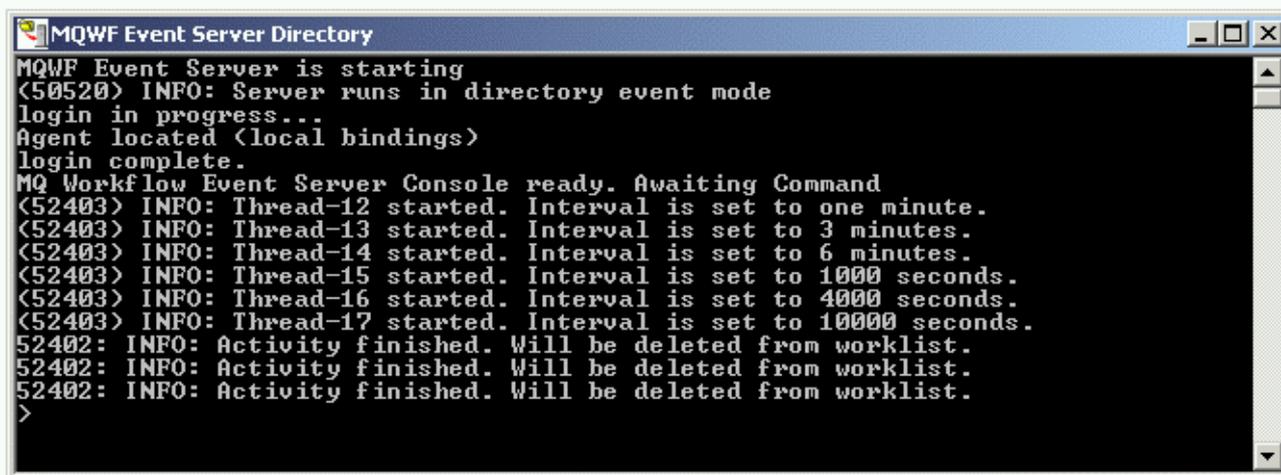
Showitems gibt die Anzahl der Workitems der aktuellen Workliste aus.

4.2.12 whatis <code>

Whatis gibt die Bedeutung eines Fehlers aus. Aufgerufen wird mit der folgenden Syntax `whatis <Fehlercode>`.

5 File Event

Das Kapitel 5 beschäftigt mit den Eigenschaften und der Funktionsweise des File Events. Das Time Event wird im darauffolgenden Kapitel ausführlich erläutert. Die Installation, Konfiguration und das Einbinden wurde bereits in den vorangegangenen Kapiteln erläutert.



```
MQWF Event Server Directory
MQWF Event Server is starting
(50520) INFO: Server runs in directory event mode
login in progress...
Agent located (local bindings)
login complete.
MQ Workflow Event Server Console ready. Awaiting Command
(52403) INFO: Thread-12 started. Interval is set to one minute.
(52403) INFO: Thread-13 started. Interval is set to 3 minutes.
(52403) INFO: Thread-14 started. Interval is set to 6 minutes.
(52403) INFO: Thread-15 started. Interval is set to 1000 seconds.
(52403) INFO: Thread-16 started. Interval is set to 4000 seconds.
(52403) INFO: Thread-17 started. Interval is set to 10000 seconds.
52402: INFO: Activity finished. Will be deleted from worklist.
52402: INFO: Activity finished. Will be deleted from worklist.
52402: INFO: Activity finished. Will be deleted from worklist.
>
```

Abbildung 22: File Event Server gestartet im Eingabeprompt

5.1 Funktionsweise

Verschiedene Verzeichnisse werden in regelmäßigen, konfigurierbaren Abständen auf Dateien überprüft. Hierzu stehen folgende Auswahlkriterien zur Verfügung:

- Das Verzeichnis ist nicht leer (beliebige Dateien)
- Eine bestimmte Datei (festgelegter Dateiname) ist vorhanden
- Eine oder mehrere Dateien mit einer festgelegten Endung sind vorhanden
- Eine oder mehrere Dateien, die einen festgelegten „Ausdruck“ im Namen enthalten

Die Suche kann mit diesen Optionen erweitert werden:

- Prüfen, ob die Datei neuer als das angegebene Datum ist (last modified date).
- Verschieben der Datei(en) in ein angegebenes Verzeichnis (mit Überschreiben oder Umbenennen einer evtl. vorhandenen Zieldatei). Falls das Zielverzeichnis nicht vorhanden ist, kann dieses erstellt werden.
- Kopieren der Datei(en) in ein angegebenes Verzeichnis (mit Überschreiben oder Umbenennen einer evtl. vorhandenen Zieldatei). Falls das Zielverzeichnis nicht vorhanden ist, kann dieses erstellt werden.
- Löschen der Datei(en)

5.2 Parameterübersicht

Parameter	Zugehöriger Wert	Erklärung
-v	Verzeichnispfad (z.B. c:\temp)	Das zu überwachende Verzeichnis
-f	Dateiname (z.B. test.jar)	Die zu suchende Datei
-p	Wert zwischen 1 - 6	Priorität wird in Sekunden Angegeben. Gibt das Intervall an, nach dem die Threads erneut durchsucht werden sollen.
-x	Gesuchte Endung (z.B. .jar)	Gesuchte letzte Zeichen der Datei (kann auch über den „.“ hinausgehen.)
-o	-	Überschreibt schon vorhandene Dateien (bei Copy und Move). Wenn nicht angegeben, werden evtl. vorhandene Zieldateien umbenannt.
-s	Enthaltener String	Teilnamensuche. Sucht in den Dateinamen nach dem angegebenen String.
-k	-	Löscht gefundene Dateien ³
-m	Zielverzeichnispfad	Verschieben von Datei(en) ³ in das angegeben Verzeichnis
-c	Zielverzeichnispfad	Kopieren von Datei(en) ³ in das angegebene Verzeichnis
-d	Datum (z.B. 25/3/2002)	Sucht nach Dateien, die neuer als das angegebene Datum sind. Zusätzlich kann noch die Uhrzeit angegeben werden (z.B. 25/3/2002/12/53/34 = 12:53:34 Uhr am 25.3.2002) Wenn keine Uhrzeit angegeben wird, ist diese standardmäßig auf 23:59:59 Uhr gesetzt
-n	-	Legt im Zusammenhang mit -m und -c ein Zielverzeichnis an (z.B. DIRECTORY -v c:\temp -p 2 -s java -m c:\temp\neu -n).

Tabelle 2: Parameterübersicht für File Event

Als Pflichtparameter müssen -v und -p angegeben werden, da diese grundsätzlich benötigt werden. Dabei handelt es sich um das zu überwachende Verzeichnis und die Priorität, mit der dieses überwacht werden soll.

5.3 Nicht kombinierbare Parameter

Einige der vorhandene Parameter („Tabelle 2: Parameterübersicht für File Event“ siehe Seite 27) sind nicht miteinander kombinierbar. Diese Kombinationen werden abgefangen und als fehlerhaft auf die Liste des Prozessadministrators gesetzt.

Diese Kombinationen sind nicht möglich:

- Kombinationen aus -m, -c und -k
- Kombination von -f mit -x
- Kombination von -f mit -s

³Dateien werden verarbeitet, wenn sie vollständig vorliegen. Dateien gelten als vollständig, wenn sie komplett vorhanden sind und nicht auf sie zugegriffen wird.

Folgende Punkte sind beim Aufruf zu beachten:

1. Am Anfang muss immer das Keyword **DIRECTORY** stehen.
2. Es muss immer das zu überwachende Verzeichnis (-v) und die Priorität (-p) angegeben werden, wobei die Prioritäten 1 – 6 zur Verfügung stehen.
3. Die Werte und ihre Parameter sind immer durch ein Leerzeichen zu trennen.
4. Lange Pfad- und Dateinamen mit enthaltenen Leerzeichen müssen von Anführungszeichen (") eingeschlossen sein (z.B. "c:\langer pfad\neu\").
5. Die erlaubten Kombination (siehe *Parameterübersicht*) müssen eingehalten werden.

5.4 Beispiele

1) **DIRECTORY -v c:\temp\ -p 1 -f javatest.txt**

Überprüft das Verzeichnis c:\temp mit Priorität 1 auf das Eintreffen der Datei „javatest.txt“

2) **DIRECTORY -v c:\temp\ -p 1 -x .txt**

Überprüft c:\temp auf das Eintreffen von einer oder mehrerer Dateien mit der Endung .txt

3) **DIRECTORY -v c:\temp\ -p 1 -s test**

Überprüft c:\temp auf das Eintreffen einer oder mehrerer Dateien, die „test“ im Dateinamen beinhalten.

4) **DIRECTORY -v c:\temp\ -p 1**

Überprüft c:\temp auf das Eintreffen irgendeiner Datei

5) **DIRECTORY -v c:\temp\ -p 1 -d 22/7/2002**

Überprüft c:\temp auf das Eintreffen einer oder mehrerer Dateien, die neuer als der 22.07.2002 sind (also last modified date ab 23.07.2002)

6) **DIRECTORY -v c:\temp\ -p 1 -d 22/7/2002/13/05/30**

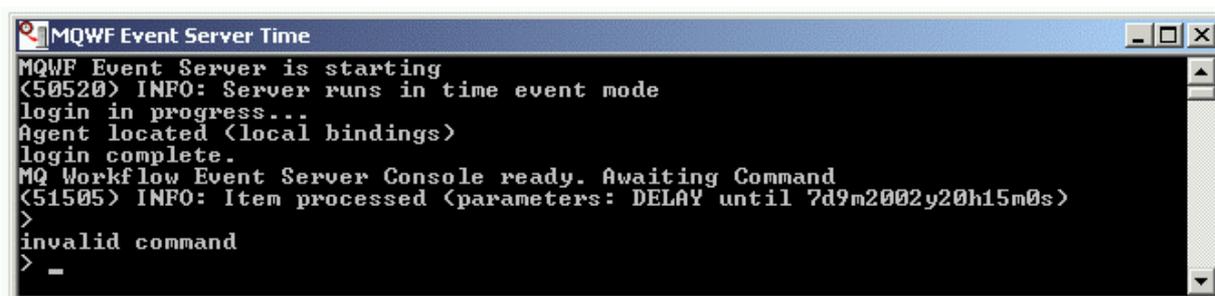
Wie 5), nur dass das last modified date ab dem 22.07.2002 um 13:05:30 Uhr gilt.

7) **DIRECTORY -v c:\temp\ -p 1 -x .txt -d 22/7/2002/13/05/30 -m c:\temp\neu**

Alle Dateien in c:\temp mit der Endung .txt, die neuer als der 22.07.2002 um 13:05:30 sind, werden in das Verzeichnis c:\temp\neu\verschoben

6 Time Event

Dieses Kapitel beschäftigt sich näher mit dem Time Event.



```

MQWF Event Server Time
MQWF Event Server is starting
(50520) INFO: Server runs in time event mode
login in progress...
Agent located (local bindings)
login complete.
MQ Workflow Event Server Console ready. Awaiting Command
(51505) INFO: Item processed (parameters: DELAY until 7d9m2002y20h15m0s)
>
invalid command
> -
  
```

Abbildung 23: Time Event Server gestartet im Eingabeprompt

6.1 Funktionsweise

Anders als das File Event, das Verzeichnisse überwacht, nimmt das Time Event auf den zeitlichen Ablauf Einfluss. Hier stehen folgende Möglichkeiten zur Verfügung:

- Eine Dauer wird festgelegt.
- Ein Zeitpunkt wird festgelegt. Dabei wird das Datum angegeben oder ein Wochentag.

Das Time Event arbeitet, ebenso wie das File Event, worklistenbasiert. In regelmäßigen Intervallen werden die Listen mit dem Server abgeglichen.

6.2 Parameterübersicht

Parameter	Zugehöriger Wert	Erklärung
for	Zeitraum	Hier wird ein Zeitraum beschrieben. Notwendige Informationen sind daher die Anzahl der Tage, Stunden Minuten und Sekunden. Syntax <Tage>D<Stunden>H<Minuten>M<Sekunden>S
until	Zeitpunkt	Bei der Beschreibung des Zeitpunkts gibt es zwei Methoden. Die erste das Angeben eines genauen Datums mit der Uhrzeit. Syntax <Tag>D<Monat>M<Jahr>Y<Stunde>H<Minute>M<Sekunde>S Die zweite ist das Angeben des Wochentags mit der erwünschten Uhrzeit. Syntax <Wochentag> <Stunde>H<Minute>M<Sekunde>S

Tabelle 3: Parameterübersicht für Time Event

Folgende Punkte sind beim Aufruf zu beachten:

1. Am Anfang muss immer das Keyword **DELAY** stehen.
2. Im Time Event stehen zwei verschiedene Parameter **FOR** und **UNTIL** zur Auswahl. Einer der Parameter muss mit dem dazugehörigen Wert dem Keyword **DELAY** folgen (siehe Tabelle 3: Parameterübersicht für Time Event Seite 29).
3. Keyword, Parameter und Wert sind durch Leerzeichen getrennt.
4. Zwischen den einzelnen Zeitwerten darf sich kein Leerzeichen befinden.

6.3 Beispiele

1) **DELAY FOR 1D2H3M40S**

Verzögerung um einen Tag, 2 Stunden, 3 Minuten und 40 Sekunden

2) **DELAY UNTIL 15D5M2002Y15H15M15S**

Verzögerung bis zum 15.5.2002 um 15.15 Uhr und 15 Sekunden

3) **DELAY UNTIL Friday 15H15M15S**

Verzögerung bis zum nächsten Freitag 15.15 Uhr und 15 Sekunden

7 Hinweise

In diesem Kapitel sind Hinweise zu Eigenschaften des Workflow Event Server zusammengefasst.

- Das Programm Workflow Event Server erlaubt nicht, dass ein Benutzer sich mehrmals anmeldet. Wird mit einem bereits aktiven Benutzer ein weiterer Event Server gestartet oder in MQSeries Workflow eingeloggt, wird die Verbindung der ersten Instanz deaktiviert, und der zuerst gestartete Event Server wird heruntergefahren.
- Mit dem Threshold kann die Größe der Workliste beschränkt werden. Hiermit wird Anzahl der Events, die zur gleichen Zeit bearbeitet werden können, eingeschränkt. Beim Ändern von Standardwerten muss bedacht werden, dass die Performance bei großen Worklisten beeinträchtigt wird.
- Das File Event gibt eine Nachricht auf der Konsole aus, wenn ein Verzeichnis mehrfach überwacht wird. Diese Nachricht ist rein informativ. Es wird empfohlen sicherzustellen, dass die File Events sich nicht gegenseitig behindern. Eine Behinderung ist möglich, wenn ein Event ein Item verschiebt oder löscht.
- Häufige Falscheingaben
 - Es wurden nicht die beiden benötigten Parameter (-v und -p) mit gültigen Werten angegeben.
DIRECTORY -x txt -p 1 -d 25/7/2002
 - Es wurde ein nicht existierendes Verzeichnis für die Parameter -v
 - Es wurde ein Leerzeichen zwischen Parameter und Wert vergessen (z.B. -p1 statt -p 1)
DIRECTORY -v „C:\Temp\“ -x txt -p1
 - Die Parameter wurden falsch kombiniert.
DIRECTORY -v „C:\Temp\“ -x txt -f „testjava.txt“ -d 25/7/2002 -p 1 -c c:\Temp\neu\

8 Beispielszenario

Folgendes Beispiel zeigt, wie der Workflow Event Server eingesetzt werden kann.

8.1 Konzeptübersicht

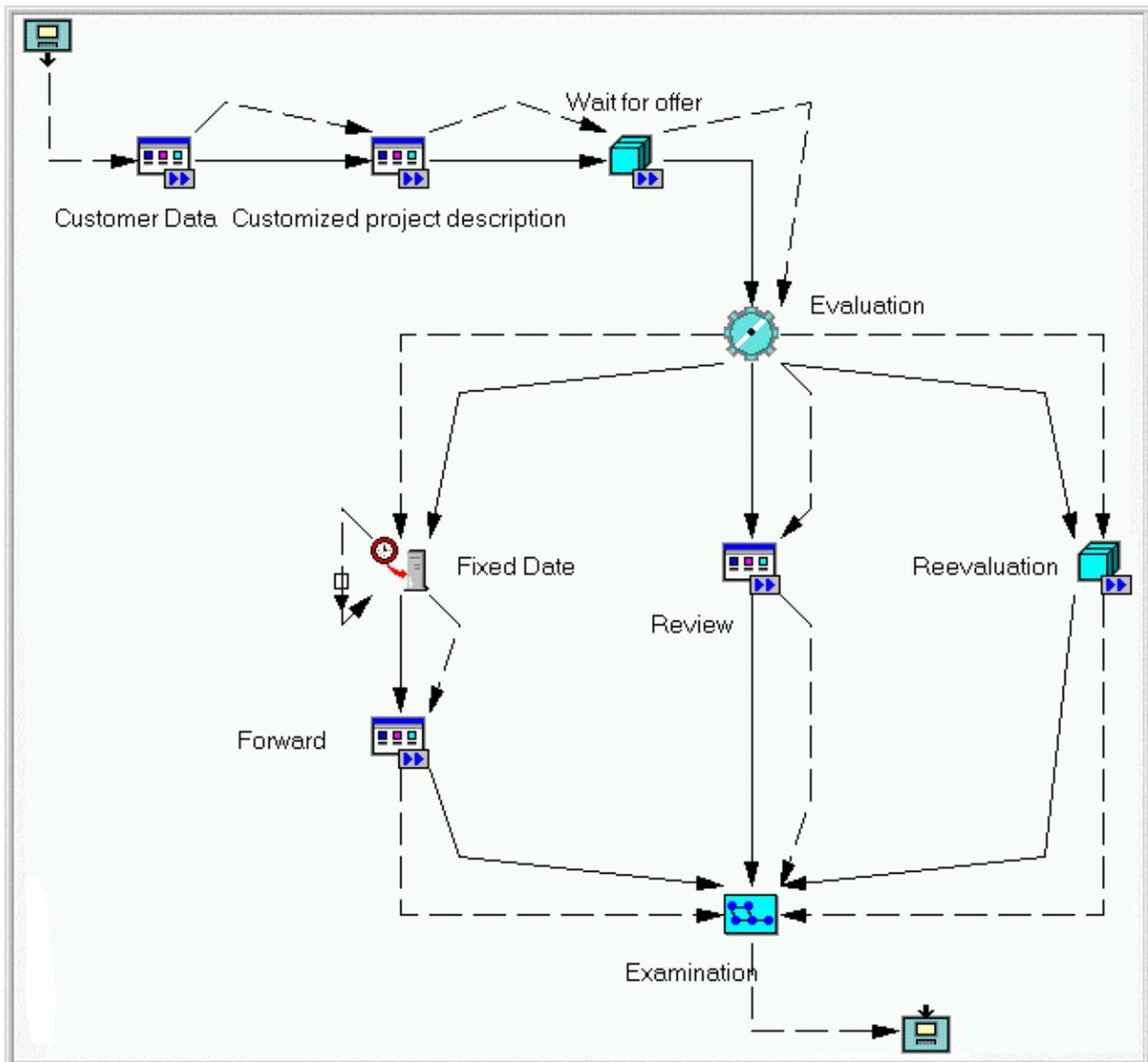


Abbildung 24: Prozessübersicht

Das Beispiel stellt einen Teilprozess einer Ausschreibung dar. Über eine Ausschreibung werden Aufträge vergeben. Teilbereiche einer Ausschreibung sind zum Beispiel die Angebotserstellung oder das Bieterverfahren. Die Angebotserstellung umfasst viele Teilschritte, die durch festgelegte Prozeduren effektiver bearbeitet werden können. Das Erstellen des Angebots folgt immer dem gleichen Prinzip (Angebotserstellungsprozess).

Im ersten Schritt wird die Ausschreibung des Auftragsgebers in das System aufgenommen und für die Firmen interne Bearbeitung aufbereitet. Nach der technischen Aufschlüsselung der Daten werden diese zur Bewertung weiter geleitet. Die Bearbeitung besteht aus mehreren Teilschritten. Durch das File Event kann das Verzeichnis überwacht werden, in dem die Dateien eintreffen werden. Damit kann sichergestellt werden, dass die Weiterführung des Prozesses erst dann erfolgt, wenn alle Informationen vollständig vorhanden sind. Das File Event achtet auf festgelegte Attribute, Dateierweiterungen, Dateinamen oder Datumswerte.

Nach dem Eintreffen der benötigten Informationen muss entschieden werden, wie mit dem Angebot weiter verfahren werden soll. Ist das Angebot bereits komplett erstellt oder ist wiederum eine Ausschreibung für ein Subunternehmer notwendig. Dabei ist nach der Komplexität und Priorität des Projektes zu entscheiden. Projekte mit einer niedrigen Priorität müssen nicht sofort gesichtet werden, somit kann die Weiterleitung auf einen bestimmten Termin verschoben werden. Mit dem Time Event kann der Zeitpunkt, an dem der Prozess fortgesetzt wird, festgelegt werden. Angebote die von hoher Priorität sind, werden sofort einen Review unterzogen. Angebote, die noch nicht vollständig sind, werden solange in einem weiteren Teilablauf (Block), überarbeitet bis alle Voraussetzungen erfüllt sind, und dann sofort weiter geleitet.

Im letzten Schritt vor dem Bieterverfahren wird das Angebot gesichtet. Die Sichtung kann wiederum mehrere Teilschritte umfassen. Diese sind aber sehr spezifisch und von der Kundensituation abhängig. Aus diesem Grund wird sie hier nur vereinfacht dargestellt. Nachdem erfolgreichen Durchlaufen des letzten Schrittes, kann nun das Angebot an den Auftraggeber übergeben werden. Somit ist der Teilprozesses des Angebotserstellens abgeschlossen.

8.2 Beschreibung der Simulation

8.2.1 Voraussetzungen

- Installation der Software wie im Kapitel 2 „Installation“ (siehe Seite 5) beschrieben
- Das mitgelieferte Programm `actimpl.exe` in `<C:\Program Files\MQSeries Workflow\BIN>` gelegt
- Die Beispiel fdl `ausschreibung.fdl`

8.2.2 Durchführung

Um das Pozeßmodell der Beispiel fdl bildhaft darzustellen, kann die `ausschreibung.fdl` in Buildtime importiert werden. Es ist wichtig das die benutzerdefinierten Icons (MQWFESF und MQWFEST) in der Datenbank bekannt sind (siehe *Einbinden von Icons* Seite 13). So können später auch Anpassungen des Testbeispiels vorgenommen werden.

Als erstes erfolgt der Import in die Runtimekomponente. Hierzu ist der folgende Befehl auszuführen:

```
C:\>fmcibie -ppassword -uADMIN -ot -i"<Pfad>\ausschreibung.fdl"
```

Jetzt kann der Client durch den Benutzer gestartet werden. Es empfiehlt sich den Workflow Event Server sowohl als Time als auch als File Event von dem Anbinden mit dem Beispiel mit den jeweils modellierten IDs zu starten und im Hintergrund laufen zu lassen.

8.2.3 Starten des Beispiels

Für dieses Beispiel wurde eine eigene Datenstruktur entworfen. In dieser Struktur sind alle wichtigen Eigenschaften für in diesem Beispiel vorkommende Dokumente abgebildet.

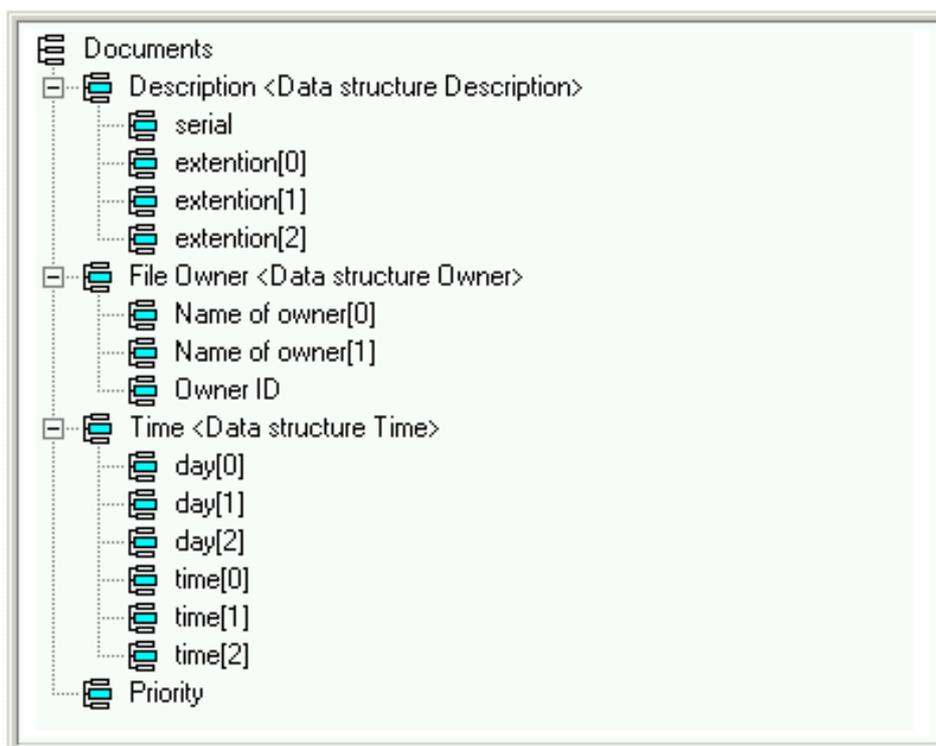
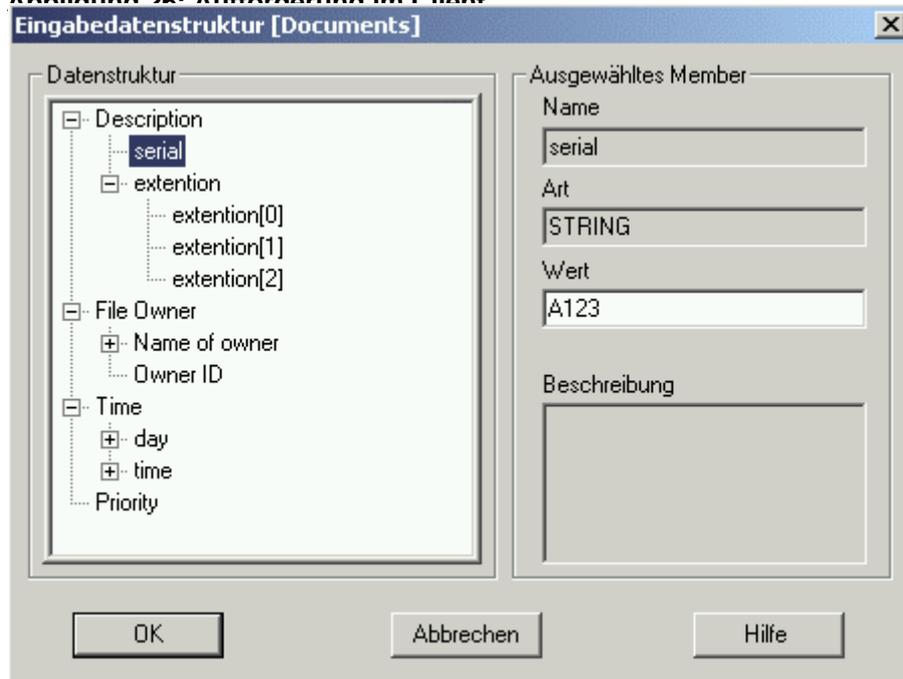


Abbildung 25: Beispiel Datenstruktur

Die wichtigen Eigenschaften in einem Dokument sind in diesem Beispiel die projektspezifische Seriennummer, die Erweiterungen für die verschiedenen Dokumente, die Beschreibung des Besitzers, eine Zeitangabe zur Abarbeitung und die Priorität.

Nach dem Starten des Prozesses wird der Benutzer gebeten, die erforderlichen Informationen in die Datenstruktur einzugeben. Die Aufforderung erfolgt mittels eines Dialogfensters (siehe *Abbildung 26: Aufforderung im Client* Seite 35), das den Namen, den Datentyp und ein Eingabefeld für den Wert angibt. Durch die Datenstruktur auf links im Fenster kann ein Member der Datenstruktur direkt ausgewählt werden.

Abbildung 26: Aufforderung im Client



Die hier gezeigte Datenstruktur ist beispielhaft.

8.2.3.1 Durchlauf Beschreibung

Nachdem die Informationen in die Datenstruktur eingetragen wurden, erscheint nun nach einem Aktualisieren der Workliste die erste Aktivität *Customer Data*. Diese muss im nächsten Schritt gestartet werden. Dieser Vorgang stellt das Erfassen der Daten des Auftraggebers und des Auftrags dar. In diesem Beispiel wird ein Programm gestartet, welches die Datenstruktur ausgibt und auffordert, die Daten für den Ausgangscontainer einzugeben *Abbildung 27: Beispielprogramm zum Ändern der Daten* (siehe Seite 35).

```

c:\program files\mqseries workflow\bin\fmctjcm.exe
Description.serial          STRING          A122
Description.extention[0]   STRING          xxx
Description.extention[1]   STRING          yyy
Description.extention[2]   STRING          zzz
File Owner.Name of owner[0] STRING          Annette
File Owner.Name of owner[1] STRING          Schneider
File Owner.Owner ID       LONG            100
Time.day[0]                LONG            28
Time.day[1]                LONG            8
Time.day[2]                LONG            2002
Time.time[0]               LONG            10
Time.time[1]               LONG            40
Time.time[2]               LONG            40
Priority                    LONG            0

Get Output Container - rc   = 0
Output Container Name      = Documents

Setting the data members...

STRING Description.serial   : A122
STRING Description.extention[0] : xxx
STRING Description.extention[1] : yyy
  
```

Abbildung 27: Beispielprogramm zum Ändern der Daten

Die zweite Aktivität *Costumized project description* stellt die firmeninterne Überarbeitung dieser Daten dar. Das Programm, das gestartet wird, ist das gleiche wie bei *Costumer Data*. Wichtig ist hier bei der Eingabe zu beachten, welche Erweiterungen und welche Serial angegeben wurde, da diese im nächsten Schritt bei dem File Event benötigt werden.



Beschreibung	Status	Art der Aktivität	Name	Prozessname	Empfangen	Eigner
Description of the job with requierments	Bereit	Programm	Customer Data	Processing the dat...	10.10.2002 20:19:...	ADMIN
Extract the information form the data fil...	Bereit	Programm	Customized project descrip...	Processing the data	10.10.2002 20:21:...	ADMIN

Abbildung 28: Ansicht der Vorgänge im Client

Als nächstes folgt das File Event. In diesem Schritt wird der Prozess solange unterbrochen, bis alle notwendigen Informationen eingetroffen sind. In diesem Beispiel wird angenommen, dass die gesamte Beschreibung in drei Dateien thematisch geordnet werden. Die Auftragsinformationen werden in technische Informationen, Auftraggeber Informationen und in eine Projektbeschreibung aufgeteilt. *Wait for offer* fasst dies in drei Aktivitäten in einem Block zusammen (siehe *Abbildung 29: Block Wait for offer* Seite 36).

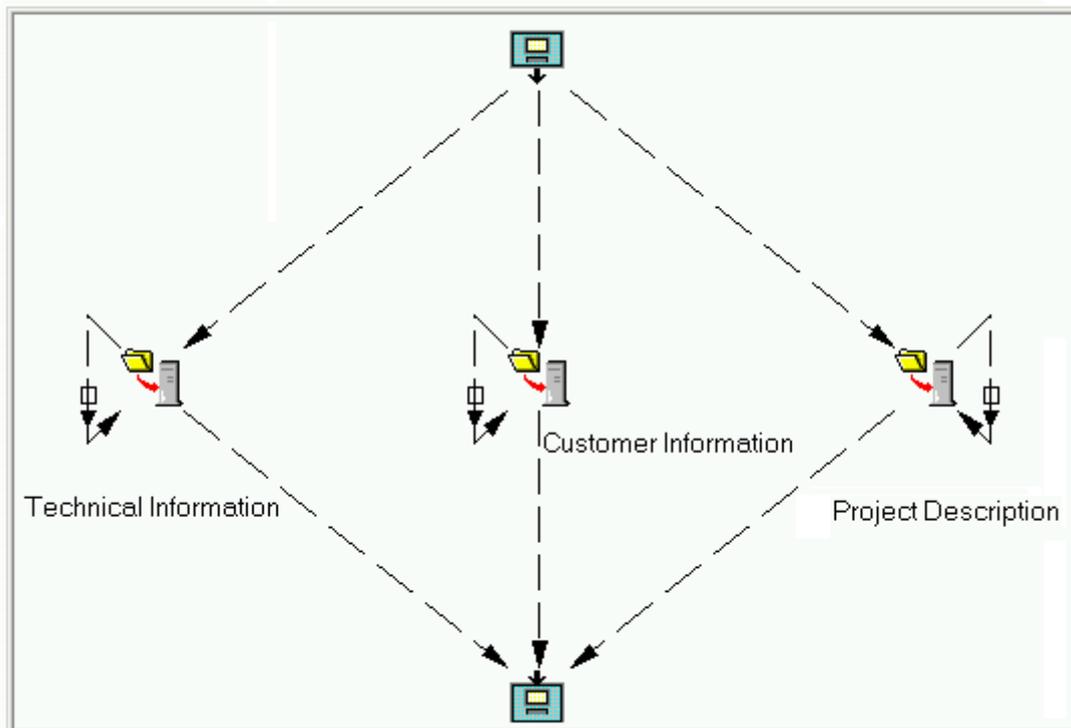


Abbildung 29: Block Wait for offer

Jede dieser Aktivitäten wartet auf eine Datei und erst wenn alle drei vollständig vorliegen, wird der Prozess fortgesetzt. Damit dieses funktioniert, ist es notwendig, dass der File Event Server läuft, und *Wait for offer* sich im „ready“ Zustand befindet (siehe *Abbildung 30: Vorgänge Wait for offer* Seite 37). Das Verzeichnispfad lautet <C:\Documents\>. Die Dateien müssen in der Name die in der Datenstruktur angegebene Serial und Erweiterung enthalten. Sobald die Dokumente in dem angegebenen Verzeichnis gefunden worden, wird *Wait for offer* beendet.



Beschreibung	Status	Art der Aktivität	Name	Prozessname	Empfangen	Eigner
Description of the job with requierments	Bereit	Programm	Customer Data	Processing the ...	10.10.2002 20:...	ADMIN
Description of the job with requierments	Bereit	Programm	Customer Data	Processing the ...	10.10.2002 20:...	FMC_DELAY
Description of the job with requierments	Bereit	Programm	Customer Data	Processing the ...	10.10.2002 20:...	FMC_DIR
DIRECTORY -v "C:\Documents" -p 1 -s A123 -x xxx	Bereit	Programm	Project Description	Processing the ...	10.10.2002 20:...	FMC_DIR
DIRECTORY -v "C:\Documents" -p 1 -s A123 -x yyy	Bereit	Programm	Customer Informati...	Processing the ...	10.10.2002 20:...	FMC_DIR
DIRECTORY -v "C:\Documents" -p 1 -s A123 -x zzz	Bereit	Programm	Technical Informat...	Processing the ...	10.10.2002 20:...	FMC_DIR

Abbildung 30: Vorgänge Wait for offer

Nun kann der Entscheidungsprozess nach der Priorität beginnen. Bei einer Ausschreibung ist es wichtig, Prioritäten zu setzen. Je nach Wichtigkeit und Komplexität des Projekts muss der Prozessablauf unterschieden werden. Projekte die von niedriger Priorität sind, können zu einem späterem Termin bearbeitet und gesichtet werden. Projekte mit hoher Priorität müssen sofort gesichtet werden und Projekte mit hoher Komplexität müssen zuerst weiter bearbeitet werden. Die Entscheidung wird in diesem Beispiel aus den Datencontainer der Aktivität *Evaluation* entnommen und auf dieser Basis entschieden.

Für den Fall der niedrigsten Priorität wird der Time Event Server eingesetzt. Der Zeitpunkt, an dem der Prozess fortgesetzt wird, wird der Datenstruktur entnommen. Der Zeitpunkt benötigt sowohl das vollständige Datum als auch eine auf Sekunden genaue Uhrzeit (siehe *Abbildung 31: Vorgang Fixed Date* Seite 37).



Beschreibung	Status	Art der Aktivität	Name	Prozessname	Programm
DELAY until 2d9m2002y19h47m50s	Bereit	Programm	Fixed Date	Processing the data	MQWFES

Abbildung 31: Vorgang Fixed Date

Sobald der Zeitpunkt erreicht ist, wird der Prozess fortgesetzt. Jetzt folgt noch das Weiterleiten und Sichten des Angebots. In Fall zwei wird das Angebot sofort weitergeleitet und im Fall drei noch bearbeitet bis das Angebot komplettiert ist und es nur noch der letzten Sichtung bedarf.

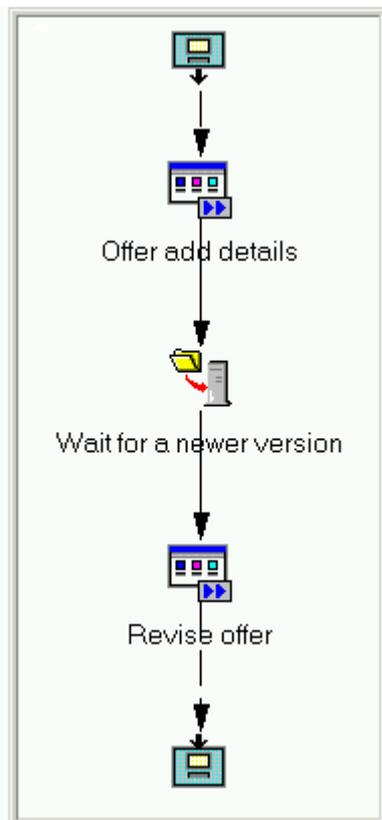


Abbildung 32: Block Reevaluation

Nach der Sichtung schließt sich die Abgabe des Angebotes an.

1. Das Angebot wird weiterbearbeitet, fehlende Details hinzugefügt.
2. Das File Event setzt den Prozess erst fort, wenn sich im Zielverzeichnis eine neuer Version der Dokumente befindet.
3. Diese Aktivität prüft ob das Angebot vollständig vorliegt. Ist es vollständig vorhanden wird der Gesamtprozess fortgesetzt, falls dieses nicht zutrifft muss das Angebot wieder vorgelegt werden.

9 Anhang

9.1 Benötigte Software

Die notwendige Software für ein Windows 2000 oder AIX System:

- **Programme:** IBM MQSeries Workflow Version 3.3.2⁴
JavaTM 2 SDK, Standard Edition Version 1.4.0
- **Softwarepaket:** mqwfes.jar, fmcojagt.jar, fmcojapi.jar

9.2 Literatur

Dieser Teil listet die empfohlene Zweitliteratur auf. In den von IBM veröffentlichten Handbüchern ist es möglich, die Funktionsweise von MQSeries Workflow in allen Aspekten genauer zu nachzulesen.

- IBM MQSeries Workflow: *Concepts and Architecture*, GH12-6285, erklärt die Grundkonzepte von MQSeries Workflow. Außerdem beschreibt es die Architektur von MQSeries Workflow und wie die Komponenten zusammenpassen.
- IBM MQSeries Workflow: *MQSeries Workflow for Windows NT for Beginners*, SG24-5848, beinhaltet die Installation und Konfiguration, den Aufbau und die Ausführung eines Geschäftsprozesses, und die Workflow API von Programmierbeispielen.
- IBM MQSeries Workflow: *Getting started with Buildtime*, SH12-6286, beschreibt die Benutzung von Buildtime von MQSeries Workflow.
- IBM MQSeries Workflow: *Getting started with Runtime*, SH12-6287, beschreibt das Starten des Clients.
- IBM MQSeries Workflow: *Installation Guide*, SH12-6288, beinhaltet Informationen und Prozeduren für die Installation und Sondereinstellungen von MQSeries Workflow
- IBM MQSeries Workflow: *Administration Guide*, SH12-6289, erklärt die Administration von MQSeries Workflow Systemen

⁴ Siehe Handbuch IBM MQSeries: Installation Guide zu den Voraussetzungen für die Betriebssysteme und Middleware.