SerCon

Service-Konzepte für Informations-Systeme GmbH

Workflow Event Server

User Manual

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

Status: **10/15/2002 7:32 PM**

Author: **Franziska Gaudeck**

Contains

1 Introduction

The Workflow Event Server is an extension of the IBM product MQSeries Workflow. MQSeries Workflow is an electronic tool to design and automate business processes. A business process describes an internal process or an internal process with external application. This process can describe parts of formalities of human recourses, purchase of material or formalities of a bidding. A process can consist of several sub processes, which again are processes by themselves, for example the writing of travelling costs, the activating of information or the writing of application. Electronic tools like MQSeries Workflow are designed to support the management to define, document, test, control, execute, improve and integrate their business processes.

The Workflow Event Server expands the possibilities of working with processes in MQSeries Workflow. It includes the Events File and Time. File offers the option to put a directory under surveillance to wait for the arrival of one or more files to copy, move, or delete files. With these options it is possible to influence the process flow through the arrival or existence of files. Time offers the possibility to continue the process after a set time, or at a specified day or weekday. This time function is designed to simplify the guidance and surveillance.

1.1 Case study

With MQSeries Workflow it is possible to describe returning processes as a network of activities. The basic idea behind the chosen example is the design of an offer (see *"Figure 1: Example for processes using File and Time Event"* page 4).

In a bidding process, information has to be collected, and then processed for company internal usage. After these tasks are completed the offer can be created. Through the usage of the File Event Servers it is possible to wait for the arrival of all files that are required for the creating of the offer. After the evaluation one, three different cases can be true.

In the first case, the offer is forwarded on a fixed date. This is possible with the functions of the Time Event. In the second case, the offer is forwarded to the final review at once. The third case describes the steps of further editing of the offer. Importance and complexity of a project determine the process flow. Before the finished offer can be presented to the customer, the offer has to be finalized on the basis of a review.

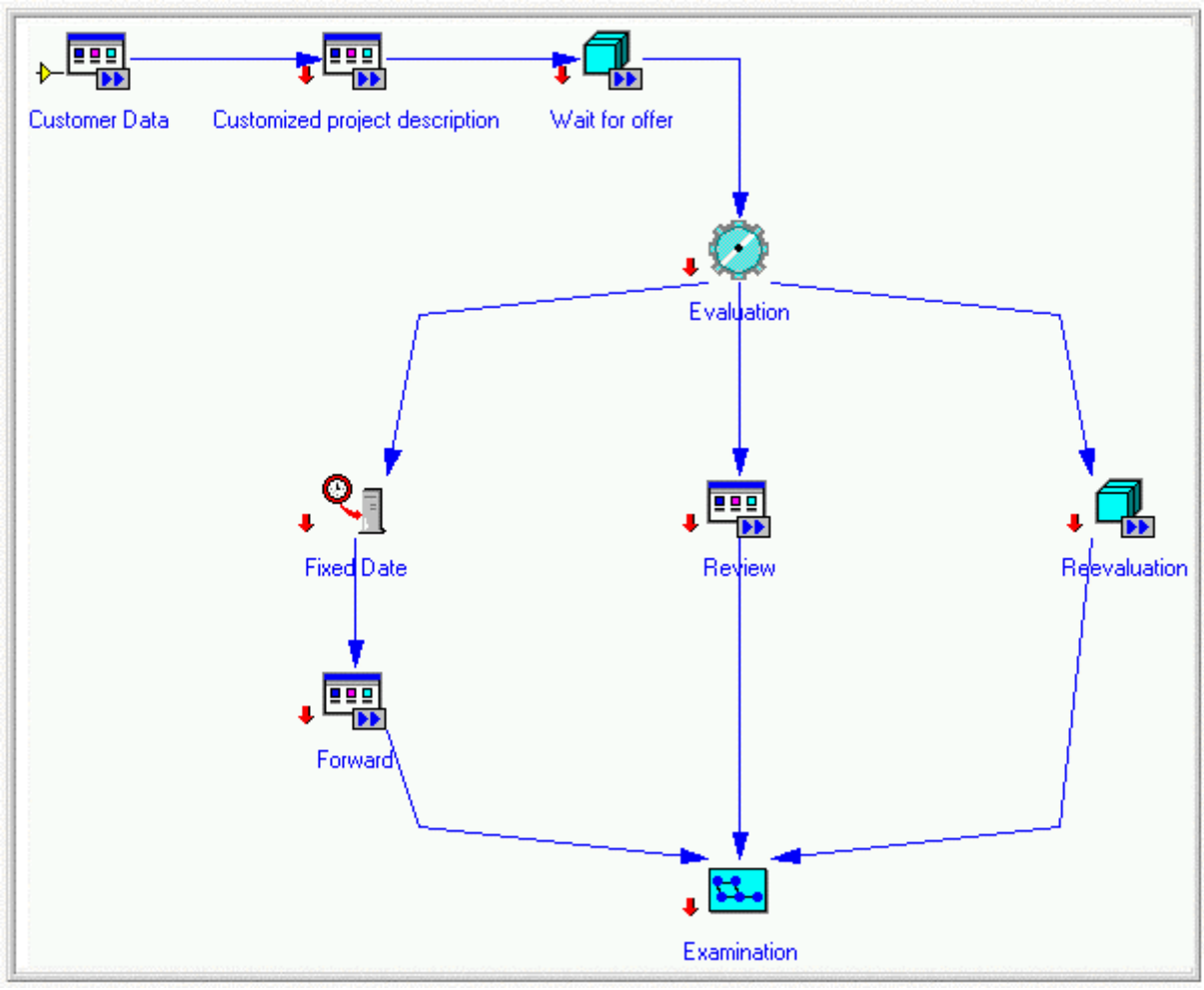


Figure 1: Example for processes using File and Time Event

2 Installation

This chapter explains the requirements, the installation and the configuration of Workflow Event Server on Microsoft Windows 2000 based systems.

2.1 Requirements

The Workflow Event Server is an extension of the IBM product MQSeries Workflow. For this reason the main requirement is an installed and configured version of MQSeries Workflow.

Furthermore a Java Runtime Environment is for the running of the Workflow Event Servers required, compatible to Sun J2SE 1.4. The Java Interpreter has to be set in the System variable %PATH%. Normally, this is done during a standard installation of Sun J2SE or J2EE automatically. If this is not the case, the path of the Java Interpreter needs to be set during every start.

Other requirements are explained in the *IBM MQSeries Workflow Installation Guide*.

2.2 Installation

The installation package contains three *.jar packages. `Mqwfes.jar` is a program package (Event Server Time and File); the other both class packages are needed for the MQSeries Workflow Java API. These two are also included in the standard package of MQSeries Workflow. These two packages need to be found the class path of the used JRE, meaning either they are

- Put in a directory that is included in the System variables %CLASSPATH%.
- Put in the directory that is set through the parameter -cp in the call of java.exe to start the Workflow Event Server (see *Starting of Workflow Event Server* page 6).
- Put in the standard class path of the JRE (`/$JAVA_HOME/jre/lib/ext`).

The last option (Standard class path SRE) is recommended.

The program package `mqwfes.jar` has to put in a local directory on the server. It is important that the path is known during the start of the server and added in the start up call (see *Starting of Workflow Event Server* page 6). This completes the installation. Furthermore the configuration and starting of the Workflow Event Server will be explained.

2.3 Starting of Workflow Event Server

The following section describes two methods to start the Workflow Event Server.

The first method is a direct call. Here, all necessary parameters are set during the start of the server.

The second method requires a configuration file, where all information is saved. In this variation all necessary data is entered and saved during the first start, and read out directly during the start.

The standard call of the Workflow Event Servers (the Java Interpreter is installed as standard and the installation package installed like recommended):

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

Is the MQSeries Workflow Java API class package not in the standard class path of the JRE the function call is the following:

```
java -jar -cp <Path of the class package> mqwfes.jar
```

The call with the basic parameters is the following:

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

2.4 Configuration

The Workflow Event Server offers two configuration possibilities.

- Setting parameters with the function call during the start

  (for Parameters see *"Table 1: Systemparameter"* page 8 and syntax see *"Starting of Workflow Event Server"* page 6). The parameters of the direct call have a higher priority than in the configuration file assigned values. The existing parameters won't be overwritten.

- Starting with the configuration file *mqwfes.ini.*

  The configuration file can be created during the start, if the parameters are not contained during the start up call. The configuration file contains all for the server important system parameters.

A line is designed in the following way:

parameter = value
(NO end of line, e.g. ';')

A line can contain key/value pairs; all that follows will be ignored. Comments can be added with ‚#'.

During a program start without parameter the existing of a configuration file is checked. If no other path is assigned through the command line the directory, in that the program package lays is searched for `mqwfes.ini`.

[1] See chapter System parameter

If the file is not found, the program offers the option to create the file, where all required values can be entered. If the possibility is not chosen, the command line is checked for parameters. If these parameters are not enough to start the server, the program will be shutdown with the appropriate error message.

2.4.1 System parameter

The table displays an overview of program parameters, their value range, default values, the keywords in the configuration file, parameter and if they can be changed with the set (explanation of the command: see *"set"* page 24).

Description in Inifile	***Value range***	***Default value***	***Parameter***	***Set[2] Command***	***Explanation***
login	String	no	-u	-	User name
password	String case-sensitive	no	-p	-	Password in plain text
service	String "DELAY", "DIRECTORY"	no	-s	-	The type of service has to set as Event.
-	boolean	true	-	Logging	Sets if the log file is written. As standard messages are written.
logsize	integer	100	-	Logsize	Logsize sets the maximal size of the log file fest. As soon it is larger, all but the last 100 entries will be deleted. The size is set in Kb. It is recommended to set the size to at least 6Kb, because 100 entries is about the size of ca. 5Kb.
Log_commands	boolean	false	-	log_commands	Writes the in the command prompt entered commands in the log file.
log_messages	boolean	false	-	log_messages	Writes the information of the application in the log file.
priority1	integer	60	-p1	priority1	Priority1 is the highest priority. It is given in seconds.
priority2	integer	300	-p2	priority2	Priority2 is the second highest priority. It is given in seconds.
priority3	integer	900	-p3	priority3	Priority3 is given in seconds.
priority4	integer	1800	-p4	priority4	Priority4 is given in seconds.
priority5	integer	3600	-p5	priority5	Priority5 is the second lowest priority. It is given in seconds.

[2] See chapter 4.2.9 „set"

Description in Inifile	***Value range***	***Default value***	***Parameter***	***Set[2] Command***	***Explanation***
priority6	integer	86400	-p6	priority6	Priority6 is the lowest priority. It is given in seconds.
refresh	integer	none	-r	refresh	Refresh sets the interval in which worklists are actualized. It is set in minutes.
logfile	Path	“mqwfes. log“	-	logfile	Log file is set through the complete path and the name of the log file.
-	Path	“mqwfes. ini“	-i	-	Inifile is set through the complete path and the name of the Inifile.
create_directories	boolean	false	-	create_directory	Creates the target directories for the File Event for die functions ”copy “ and “move“ automatically, if it does not exist.
threshold	integer	300	-	threshold	Threshold sets the number of Events contained in the worklist.
tracing	boolean	false	-	tracing	Tracing writes all Information and errors in the log file together with exact timestamp

Table 1: Systemparameter

Some of the system parameter (see *“Table 1: Systemparameter”*) can be changed set command during the running, though the values in the configuration file won’t be changed.

3 Modeling in MQSeries Workflow

After the Installation of Workflow Event Servers was described in the previous chapter, this chapter explains the implementing of the File and Time Event into MQSeries Workflow with the help of the application Buildtime.

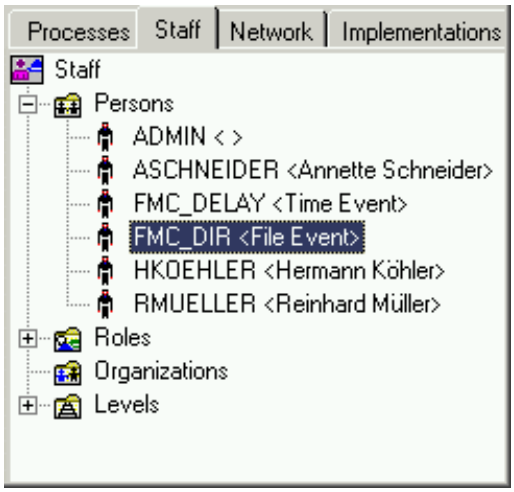
3.1 Workflow Configuration

Before the Workflow Event Server can be integrated in a Workflow system, the following configurations have to be done in Buildtime.

- Creating of the Users: One user is needed for each Event server
- Setting of the rights ”operation administration“ for the appropriate Category and ”Persons: Workitem“ ? ”all users“
- Optional: Adding of Icons for the Events File and Time

3.1.1 Example for the Event File

The following section explains the necessary settings for the example of the File Event, the settings for the Time Event are identical. In the example the Time Event is called ”FMC_DELAY“ and the File Event ”FMC_DIR“.



The first step is the creating of a new user in Buildtime under ”Staff“ with a right mouse click on ”Persons“,”new Person“.

Figure 2: Creating of a User

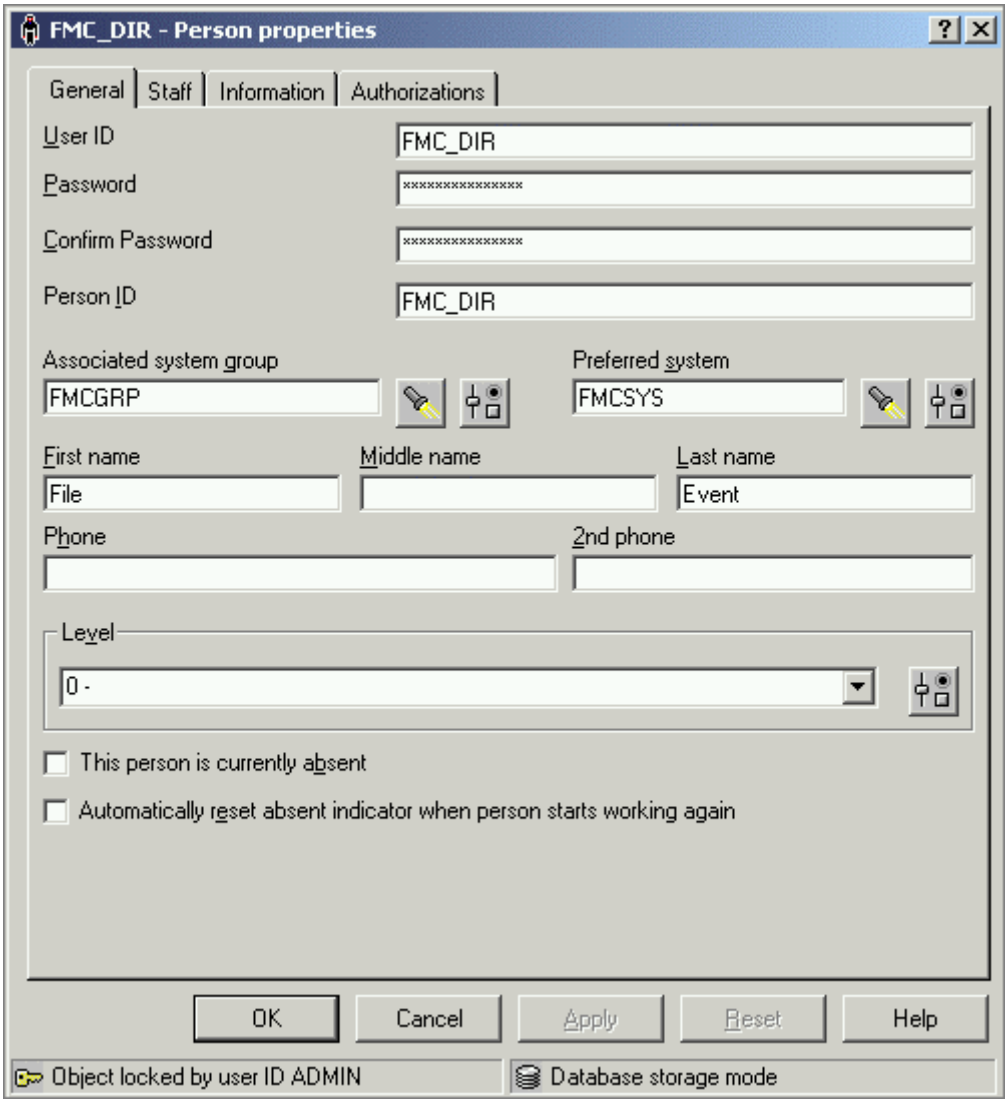


Figure 3: Entering of user information

The user information is entered under ”Properties“. For the properties of the User ID and the password, it is important to ensure the parameters are equal to the parameters in the `mqwfes.ini`.

3.1.2 Rights management

The user requires special rights access the Server.

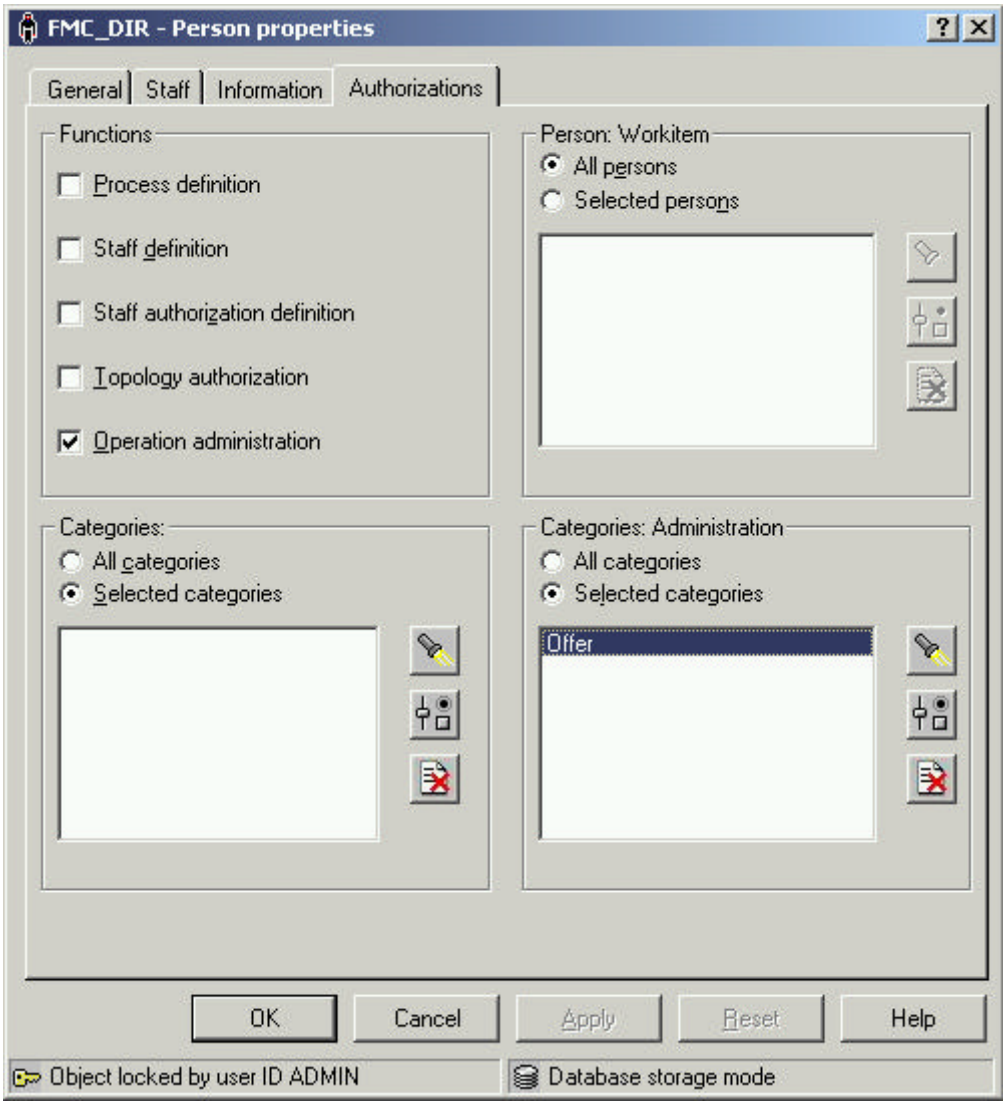


Figure 4: Setting of rights

In the properties of the users under the point ”Authorizations“ the right ”Operation administration“ has to be given.

Further the rights under ”Workitem“ have to be issued for ”All persons“ and under ”Administration“ in which the appropriate categories have to be set.

3.2 Workflow Design

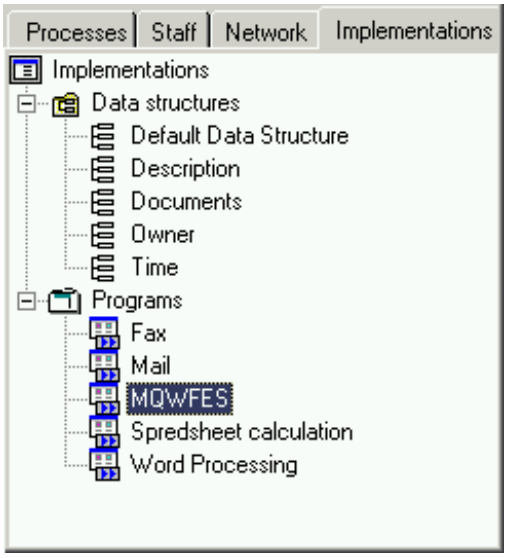
To use the functionality of the Workflow Event Server in a Workflow process, an activity needs to be created in the Buildtime at the specific location of the process. The creation of activities is also explained in *MQ Workflow Concepts and Architecture* and *Getting Started with Buildtime*.

The Event activity requires the following properties:

- Start manual
- Exit automatic
- The user of the Workflow Event Server has to be set explicit under “Staff” (or in case the item is transferred while running the process the process administrator)
- In the description field Event properties have to be set, this means **DELAY** or **DIRECTORY** have to keyword together with the parameters.
- A implemented activity program

3.2.1 Creating of the program

The first step in the design of the process in Buildtime is the creation of a program. This program is needed for running the Event. Because the program will not be started in the current implementation, it is not important which application is used.



A new program is created under “Implementations” “Programs”, in this example it is called MQWFES.

Figure 5: Creating of the program

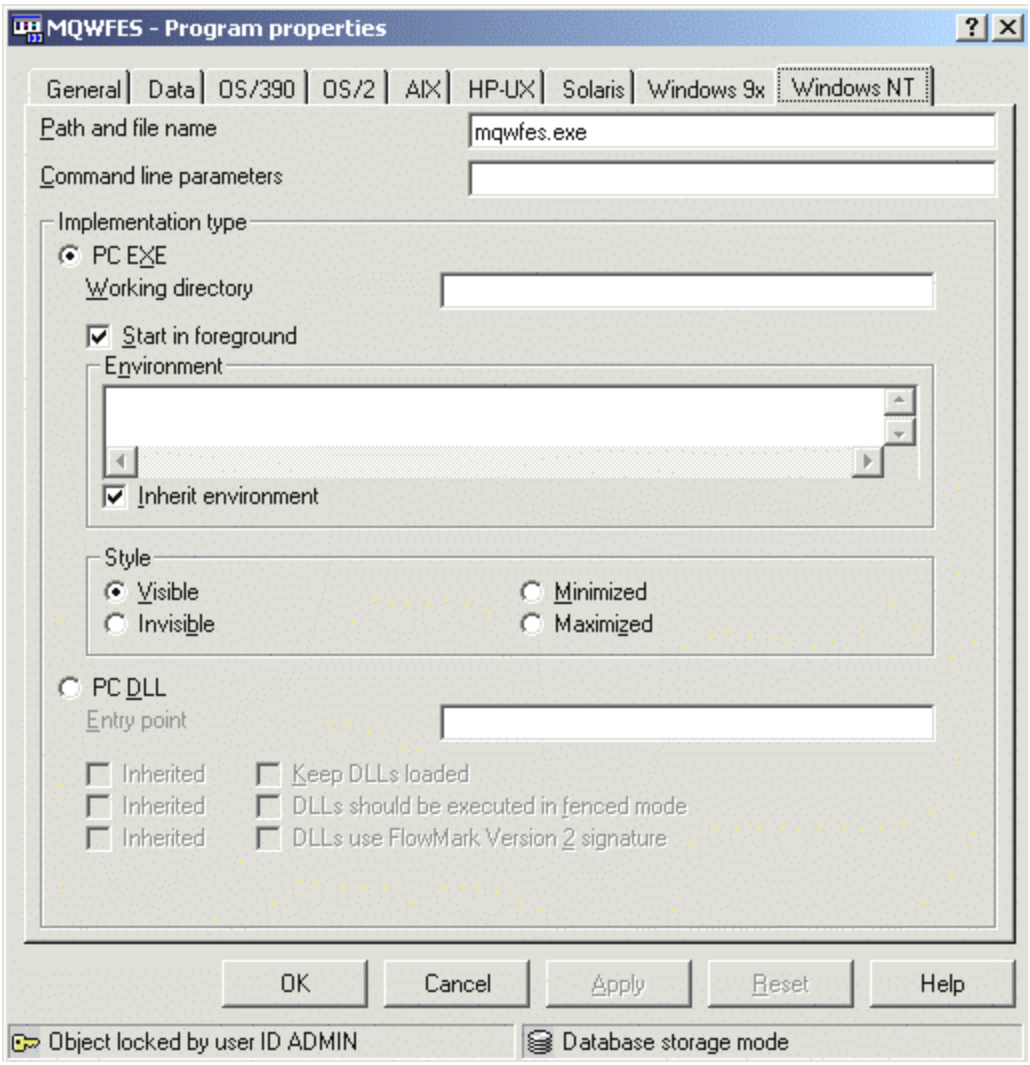


Figure 6: Setting of properties

With this example it is shown that a syntactically correct (but not necessary existing) application has to be entered in the properties.

Furthermore, the data structures have to be checked that they correspond to the process. The simplest method to ensure this is to make the program universally usable set under the point Data *"Program can handle any data structures*". Otherwise the selected data structure has to be the same as used in the container.

3.2.2 Importing of Icons

The Icons used in the Events File and Time Icons are included program package. To be able to use them, they need to be imported in the Buildtime database. The importing in the Buildtime database is realized with Buildtime.

These are the necessary steps:

1. Creating of a new or opening an existing process

2. Activate Process Diagram Window

3. Choose the point *Customize* that can be found under *View*, *Drawing Tools* (see *"Figure 7: Selecting of Drawing Tool Sets"* page 14)

4. Add the icons with *Add* (see *"Figure 8: Drawing Tool Sets (1)"* page 15)

5. Move with the help of Drag & Drop auf *"Additional Icons for this Activity Type*" to "*Program Activity*" (see *Figure 9: Drawing Tool Sets (2)* page 15)

6. Affirm with OK (see *Figure 10: Added Icons* page 16)

To use the example shown icons, they have to be imported in the database with the names MQWFESF and MQWFEST.

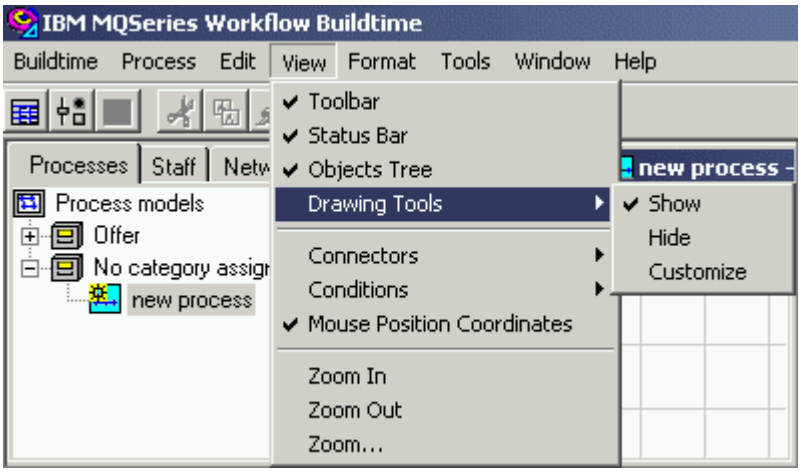


Figure 7: Selecting of Drawing Tool Sets

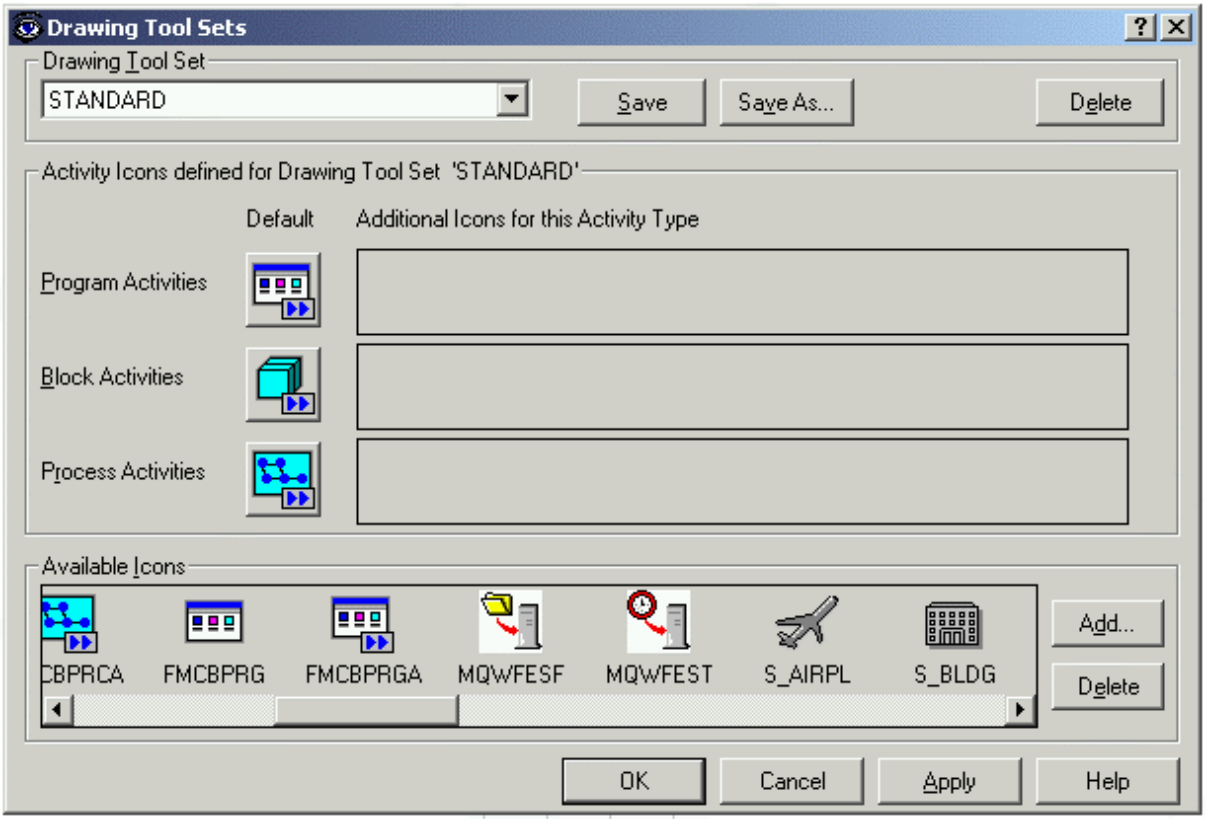


Figure 8: Drawing Tool Sets (1)

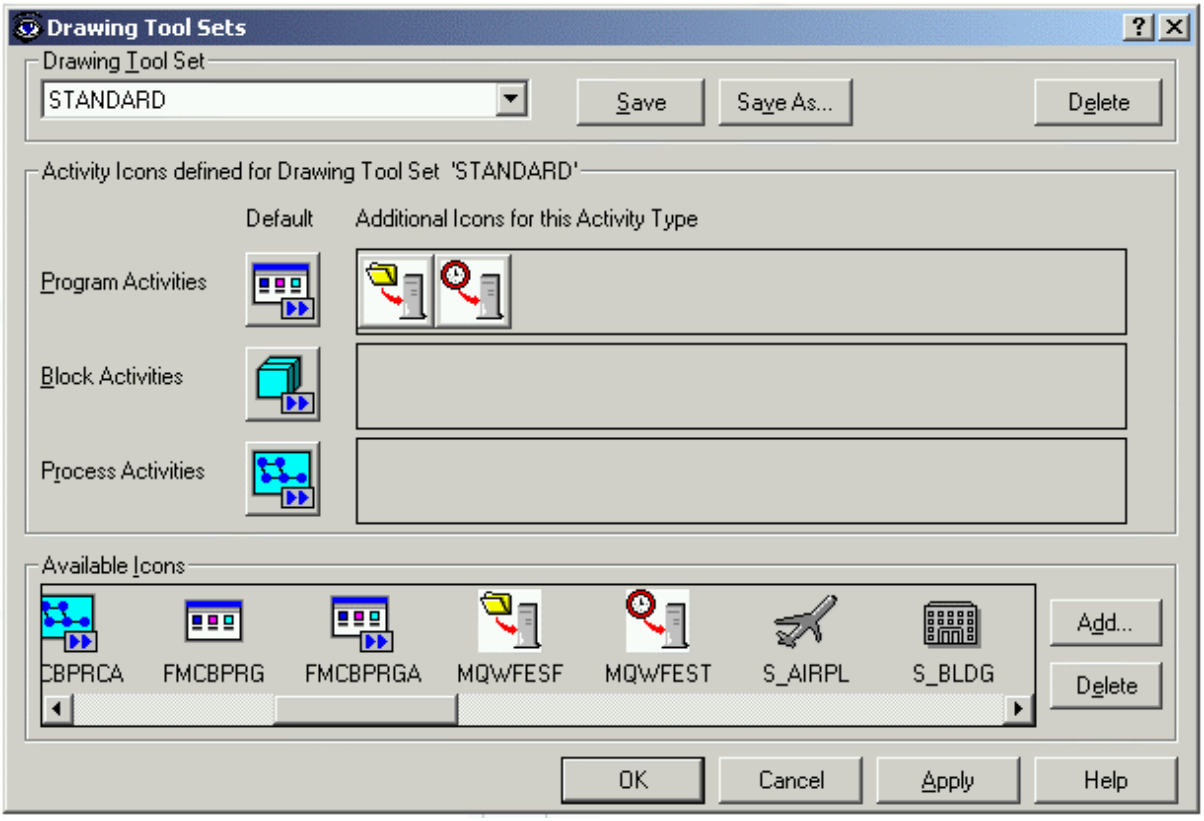


Figure 9: Drawing Tool Sets (2)

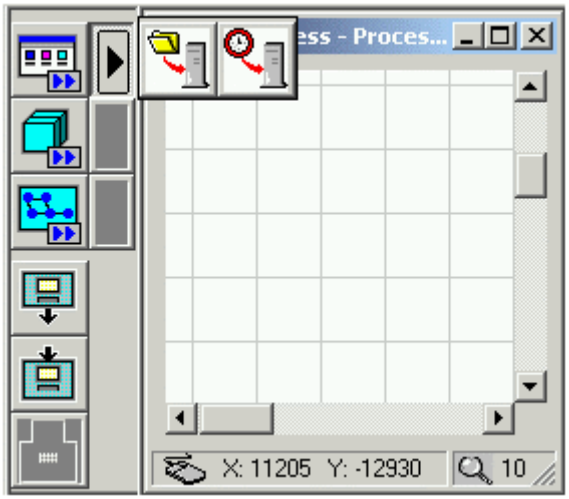


Figure 10: Added Icons

3.2.3 Creating of a Activity

In the process a normal program activity is created.

In the properties of the activity the required parameters have be added under ”General“. It needs to be observed that the keyword (`DIRECTORY` or `DELAY`) stands in the beginning of the description field, this has to be the same as in the configuration file *mqwfes.ini*. If the description does not start with the right keyword, the activity will be ignored by the Event Server.

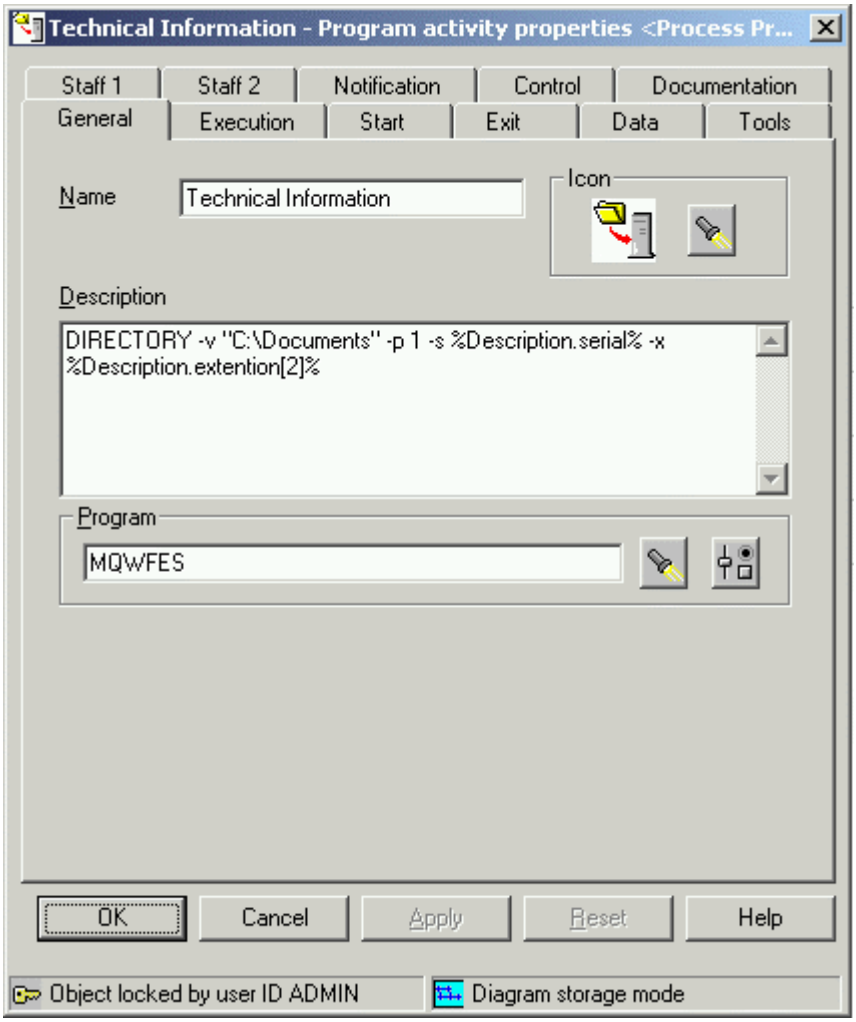


Figure 11: Attributes of the Activity (1)

Parameter values can be read from a data structure. In the example see the description field *"Figure 11: Attributes of the Activity* (1)" the information is read from the underlying data structure. This is accomplished with place holders. The field from the data structure is addressed simply by setting the path and enclose it with ‚%'. (More Information about setting of parameters can be found in*6.2 Parameter*" page 27)

The name of the program can be chosen freely.

The under section *"Creating of the* program*"* page 12 created program has to be selected under *"Program*".

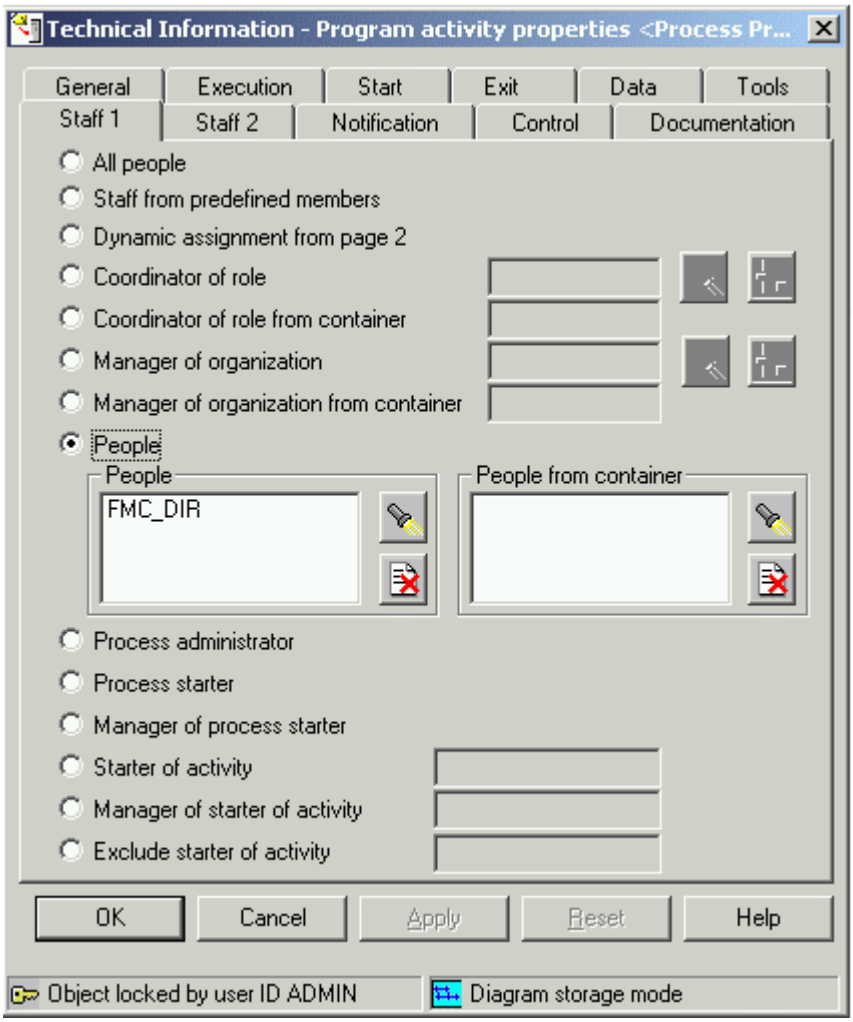


Figure 12: Attributes of the Activity (2)

Only for the Event created user might be chosen under the "*People*" in "*Staff 1*"(see *"Figure 12: Attributes of the Activity (2)"* page 18).

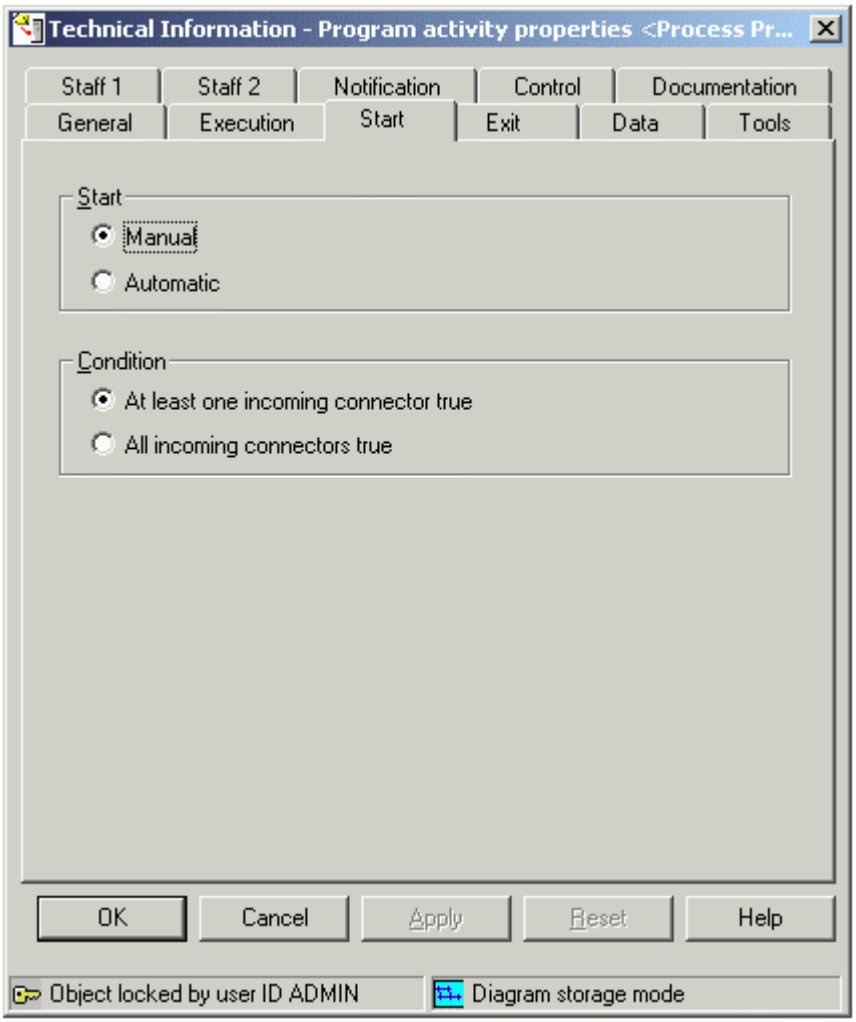


Figure 13: Attributes of the Activity (3)

As a last configuration “starting of the activity” needs to be set on “Manual“.

3.2.4 Passing of data structures

With the help of data structures it is possible to transfer Information between the activities (see chapter **Fehler! Verweisquelle konnte nicht gefunden werden.** ”***Fehler! Verweisquelle konnte nicht gefunden werden.***“ page **Fehler! Textmarke nicht definiert.**). Only when an activity is started data structures can be passed actively.

The Workflow Event Server does not start an activity while processing an Event. For transmitting of data, reading from the input container and writing into the output container the Data Default Connecter is used.

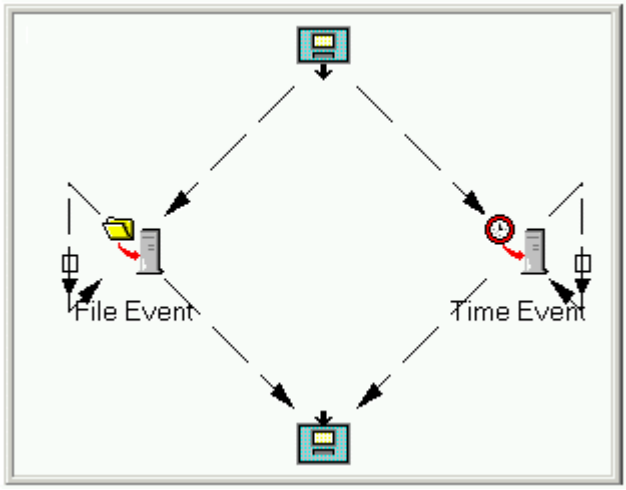


Figure 14: data connectors

4 Possibilities of controlling by the Administrator

The Administrator has two possibilities to influence the process flow. As process administrator has the Administrator the possibility to transfer items to worklists of the Event Server. In the command prompt he is also able to change properties of items. With these options the Administrator is able correct invalid items and ensures a continuing process flow. This chapter is going to explain these two options closer.

4.1 Administration in the Client

The in Buildtime designed process is started in the Client. The used Client is the included standard Client. Often special masks are created for company intern applications, but the basic functionality is the same. This section describes the possibilities of controlling in the Client (more information can be found in *Getting started with Runtime*).

4.1.1 Import into Runtime

Before the in Buildtime designed process can be started in the Client, the process has to be imported in the Runtime component. The import is realized with the following call:

```
C:\>fmcibie -ppassword -uADMIN -ot -i"<path>\<name>.fdl"
```

Now the process can be started in the Client. Detailed Information about the import can be found in the redbook *MQSeries Workflow: Getting started with Runtime*.
User defined Icons have to be known in the system before the process can be imported.

4.1.2 Working in the Client

Before the user is able to start the processes, the process administrator has made some configurations. The standard process administrator is defined as user ”Admin“ with ”password“ a password. In window *tree structure overview* the process is displayed in three different levels.

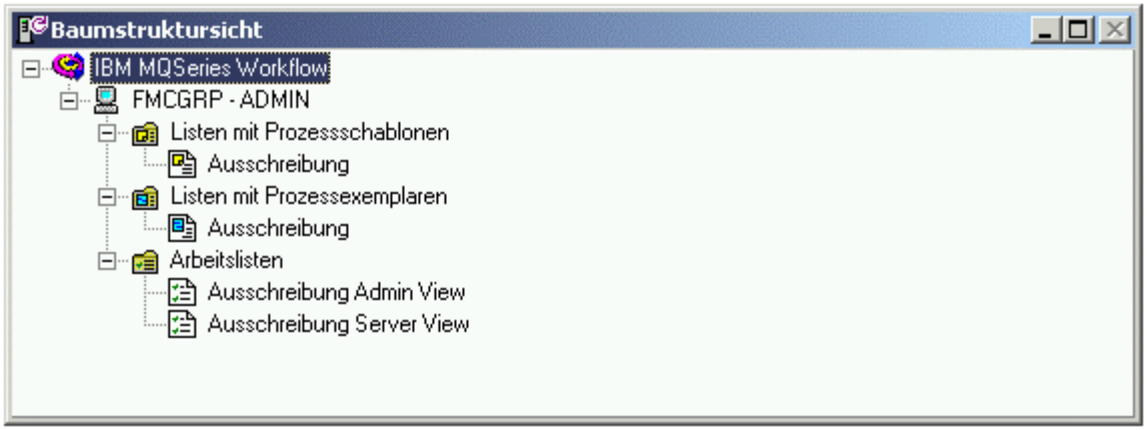


Figure 15: tree structure in the Client
The first step for the process administrator is to create a list of process templates, a list of process instances and a worklist in the window *tree structure*.

Figure 16: process template with processes

In the process template process instances can be created multiple times. All running processes can be found in the list of process template.

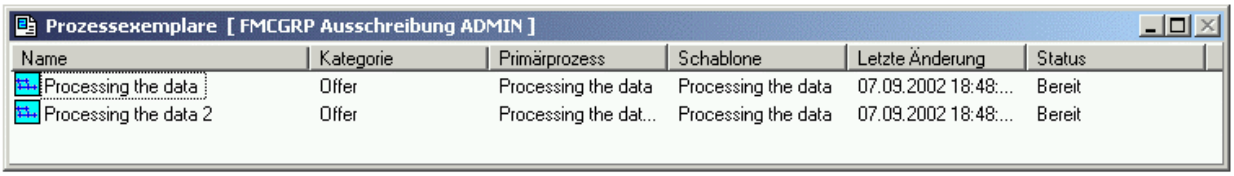


Figure 17: Process instance processing the data

In the worklist (Vorgänge) all current activities are displayed. In this window the status of the activities can be checked and changed. An activity can be started, restarted, but also finished.

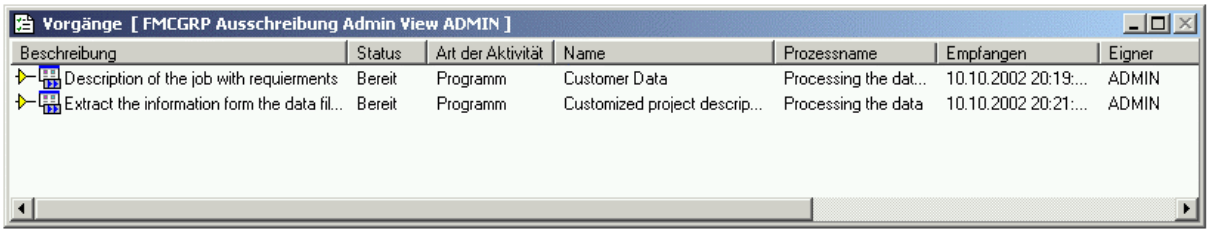


Figure 18: Worklist with Activities

4.1.3 Error handling

This section describes the possibilities of the process administrator to handle invalid activities.

4.1.3.1 Transfer of a process by the Administrator

If the Time or File Event in syntactically incorrect, the item will be moved on the worklist of the process administrator. Is an item on the worklist of the process administrator he has the possibility to change the description field of the item to correct the error. To change the description field open the properties of the activity (see context menu open with the help of the right mouse click). After the change is applied the item can be moved to the worklist of the appropriate Event Server. Is syntax correct but the contents wrong, the changes can be done in the command prompt (see 4.2.2 ”*change*” page 23).

4.1.3.2 Forced finish

If an activity can not be finished in a normal manner, the option to forced finish has to be used.

4.2 Commands for the prompt

The Event Server is started in a command prompt. In the prompt window all important status messages are displayed, details are written directly in the log file `mqwfes.log`. With the help of commands all information can be displayed in the command prompt. This section explains the available commands..

4.2.1 "?" or help

With ? or `help` all commands can be listed. All commands will be displayed on the command prompt together with a short description. With this option the navigation should be simplified.

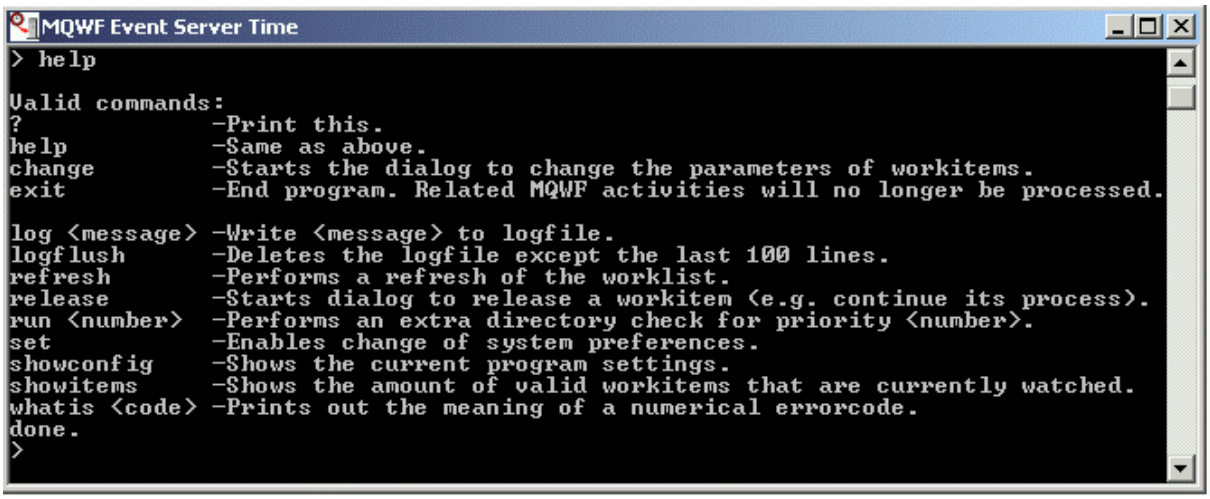


Figure 19: displays help in command prompt

4.2.2 change

`Change` starts a dialog, to change the parameters of a workitems. The category, process instance, name of the activity and the new parameter will be requested. All available options will be displayed.

4.2.3 exit

`Exit` is needed to shutdown the Event Server. Processing of the events is stopped.

4.2.4 log <message>

This command is used to write messages in the log file. It can be useful to find special entries in the log file.

The syntax is: `log <message>`.

`<message>` in the entered text, it is finished with "Enter".

4.2.5 logflush

The log file has a specified size. The standard value is 100KB. This value can be changed by the administrator. When the file is larger than the set file size, all but the last 100 lines will be deleted. This function can be started in the command prompt with the command `<logflush>` manually.

4.2.6 refresh

With `refresh` all current worklists are acquired from the MQSeries Workflow Runtime Server independent of the set refresh rate.

4.2.7 release

`Release` starts a dialog to process a specific workitem manually. The corresponding process will be continued and the workitem won't reappear in the worklist.

4.2.8 run <number>

Run initialized the manual checking of Directory Items with the selected priority. All priorities will be checked if the value 0 for `<number>` is entered.

4.2.9 set

The command `set` offers the possibility, to change the program parameter (see "*Table 1: Systemparameter*" page 8). The changes won't be applied in the configuration file. The changes can be initialized with the help of a dialog that can be started with `set`. Or it can be changed directly with `set <parameter>` and value is entered with a dialog or with `set <parameter> <value>`. With entering of `set`, all system parameter will be listed.

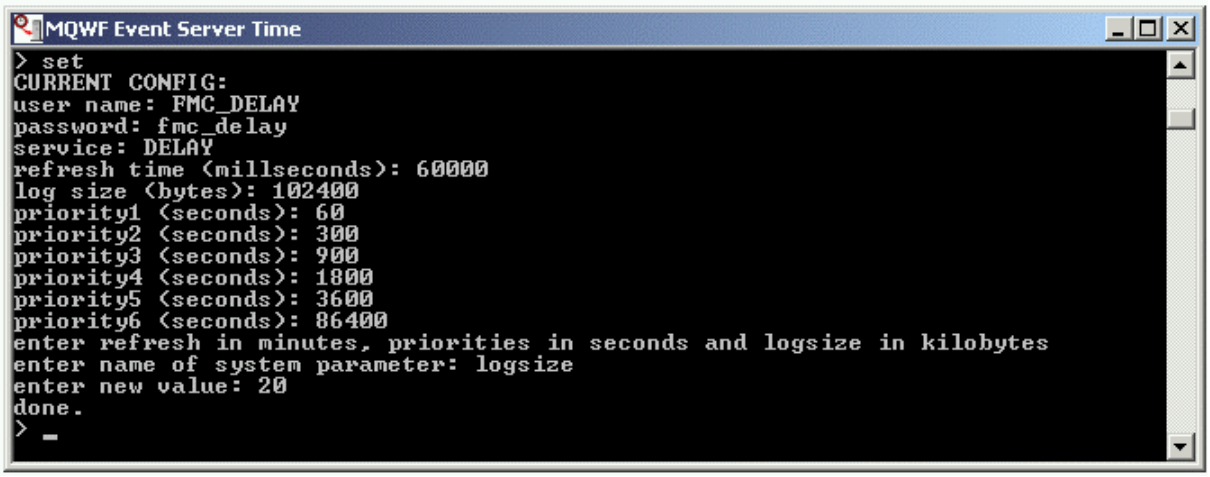


Figure 20: set dialog in a command prompt

4.2.10 showconfig

Showconfig displays the complete configuration of the program. From the displayed parameters refresh, logsize and the priorities can be changed (see *"set"*).

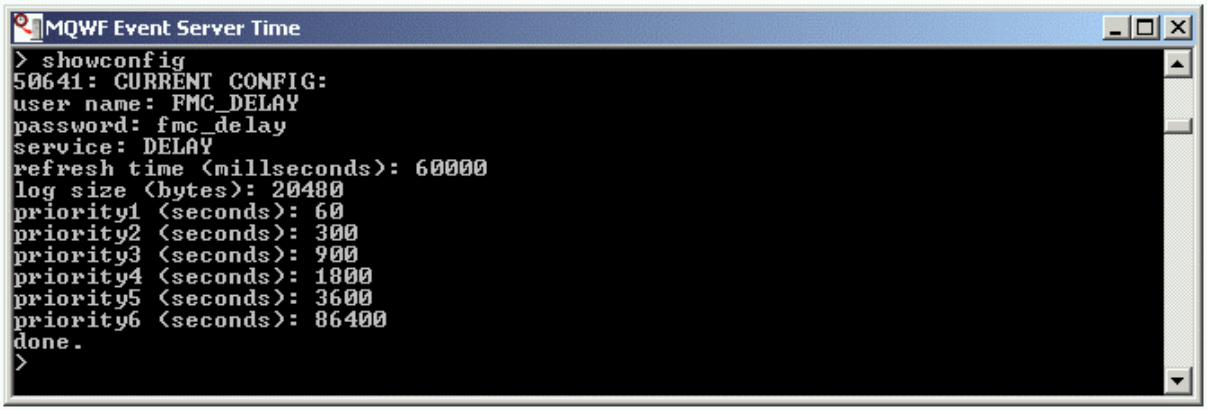


Figure 21: showconfig in a command prompt

4.2.11 showitems

Showitems displays the number of workitems in the current worklist.

4.2.12 whatis <code>

Whatis displays the meaning of an error in the command prompt. The syntax of the call is whatis <Error code>.

5 File Event

Chapter 5 explains the properties and functioning of the File Events. The Time Event will be described in the next chapter. The installation, configuration and the implementation were already explained in the previous chapters.

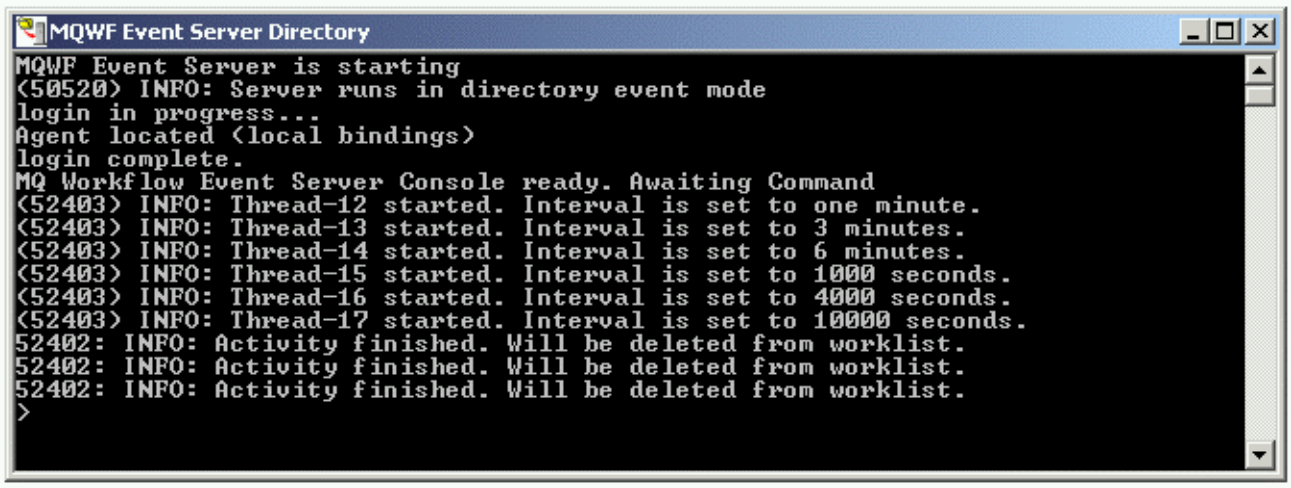


Figure 22: Starting the File Event in the command prompt

5.1 Functioning

Different directories are checked for files in regularly configured intervals. The following criteria are available:

- The directory is not empty (any files)
- A specified file (specified filename)exists
- One or more files with a specified extension exist
- One or more files that contain a specified expression in the name

The search can be extended with the following options:

- Checking if the file is newer than the given date (last modified date).
- Moving of files in a specified directory (with overwriting or renaming of possible existing files in the target directory). In case the target directory does not exit it can be created.
- Copying of files in a specified target directory (with overwriting or renaming of possible existing files in the target directory). In case the target directory does not exit it can be created.
- Deleting of files

5.2 Parameter overview

Parameter	*Value*	*Explanations*
-v	Directory path (e.g. c:\temp\)	The directory that is put under surveillance.
-f	File name (e.g. test.jar)	The file that is searched for.
-p	Value between 1 - 6	Priority is set in seconds. Sets the interval that gives the time frame after the thread is searched again.
-x	Extension (e.g. .jar)	Compares the last characters in the file name (can exceed ".").
-o	-	Overwrites already existing files (only copy and move). If it is not set, possible existing files will be renamed.
-s	Contained String	Searching parts of the file name. Searching in the file name for the given string.
-k	-	Deletes the found files[3].
-m	Target directory	Moving of files[3] in the specified target directory.
-c	Target directory	Coping of files[3] in the specified target directory.
-d	Date (e.g. 25/3/2002)	Searches for files, that are newer than the specified date. In addition the time can be set two (e.g. 25/3/2002/12/53/34 = 12:53:34 on 25.3.2002) If the time is not set the standard value is 23:59:59.
-n	-	Creates in the connection with –m and –c a target directory (e.g. DIRECTORY -v c:\temp\ -p 2 -s java -m c:\temp\neu\ -n).

Table 2: Parameter Overview for the File Event

The required parameters are –v and –p. With these two parameters, the directory path and the priority are set.

5.3 Parameters that can't be combined

Some of the parameters (see "*Table 2: Parameter Overview for the File Event*" page 27") can not be combined. These combinations will be caught and will be set as incorrect on the list of the process administrator.

The following combinations are not possible:

- combinations of –m, –c and –k
- combination of –f with –x
- combination of –f with –s

[3]Files will be processed if they are complete. Files are complete, if they exist completely and are not accessed by an other applications.

The following points have to be observed in a function call:

1. The call has to start with the keyword DIRECTORY.
2. The path for the directory (-v) that is put under surveillance and the priority (-p) have to be set. Priorities 1 – 6 are available.
3. The value and their parameter are separate by a white space.
4. Long path and file names that contain blanks have to be put in quotation marks (") (e.g. `"c:\long path\new\`)
5. Only valid combinations (see *Parameter Overview for the File Event*) may be used.

5.4 Examples

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

Check the directory c:\temp with the priority 1 for the file „javatest.txt“

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

Check c:\temp for the existing of one or more files with the extension .txt

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

Check c:\temp\ for the existing of one or more files, which contain „test“ in the file name.

4) **`DIRECTORY –v c:\temp\–p 1`**

Check c:\temp\ for the existing of any files

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

Check c:\temp\ for the existing of one or more files, are newer than 22.07.2002 (also last modified date starting at 23.07.2002)

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

Like 5), only that the last modified date starts on 22.07.2002 13:05:30.

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

All files in c:\temp\ with the extension .txt, that are newer than 22.07.2002 13:05:30, are moved to the target directory c:\temp\neu\

6 Time Event

This chapter explains the Time Event closer.

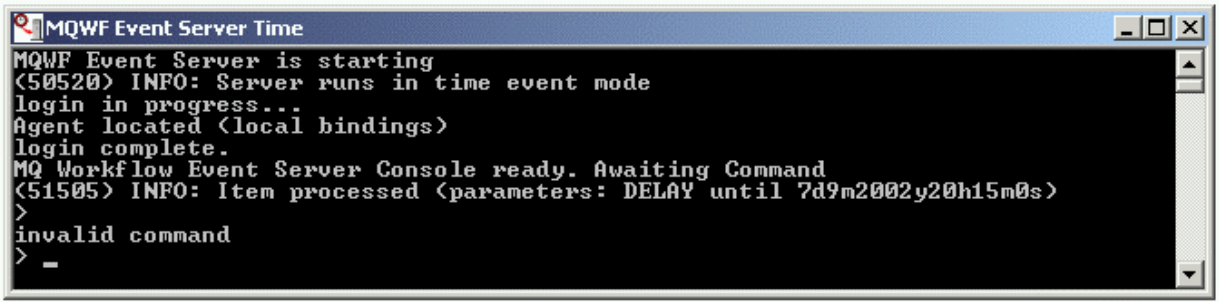


Figure 23: Starting of Time Event Server in command prompt

6.1 Functioning

While the File Event puts directories under surveillance, the Time Event influences the time flow.

Here the following options are:

- Duration is fixed.
- A point in time is fixed. Here either a date or weekday is set.

The Time functions like the File Event worklist based. The lists will be compared with the ones on the server in regular intervals.

6.2 Parameter overview

Parameter	***Value***	***Explanation***
for	Duration	Here the duration is going to be described. The necessary information is given in the number of days, minutes and seconds. Syntax `<Days>D<Hours>H<Minutes>M<Seconds>S`

until	Fixed point in time	For the description of a point in time two methods exist.
		
		The first is to set an exact date with the time.
		
		Syntax
		`<Tag>D<Months>M<Year>Y<Hours>H<Minutes>M<Seconds>S`
		
		The second is to set a weekday with the time.
		
		Syntax
		`<Weekdays> <Hours>H<Minutes>M<Seconds>S`

Table 3: Parameter Overview for the Time Event

The following points have to be observed in a function call:

1. The call has to start with the keyword DELAY.
2. In the Time Event it can be chosen between the two different parameters FOR and UNTIL. One of the parameters and the connected value has to follow the keyword DELAY (see *"Table 3: Parameter Overview for the Time Event"* page 29).
3. Keyword, parameter and value have to be separated by white space.
4. Between the single parts of the value no white space is allowed.

6.3 Examples

1) **`DELAY FOR 1D2H3M40S`**

Delay for one day, two hours, three minutes and forty seconds

2) **`DELAY UNTIL 15D5M2002Y15H15M15S`**

Delay till 5/15/2002 at 3.15 pm and 15 seconds

3) **`DELAY UNTIL Friday 15H15M15S`**

Delay till the next Fridays at 3.15 pm and 15 seconds

7 Notices

This chapter contains some notices about properties of the Workflow Event Server.

- The application Workflow Event Server does not offer the possibility to connect one user more then once. If a server with an already active user started or logged in MQSeries Workflow, the connection of the first instance will be deactivated and this server will be shutdown.

- With Threshold it is possible to set the maximal size of a worklist. So it is possible to limit the number of Events that can be processed simultaneously. When changing the standard values it has to be considered that large worklists influence the performance.

- The File Event displays a message if a directory is under surveillance multiple times. This message is purely informative. It is recommended to check so that Events won't influence each other. Influencing is possible if an Event moves or deletes a file.

- Usual invalid entries

  - Not both required parameters (-v and –p) with valid values were set.
    ```
    DIRECTORY –x txt –p 1 –d 25/7/2002
    ```
  - A not existing path for the directory (-v) was set
  - A white space was forgotten between parameter and value (e.g. –p1 instate of –p 1)
    ```
    DIRECTORY –v ”C:\Temp\“ –x txt –p1
    ```
  - Invalid combination of parameters.
    ```
    DIRECTORY -v ”C:\Temp\“ -x txt -f ”testjava.txt“ -d
    25/7/2002 -p 1 -c c:\Temp\neu\
    ```

8 Case study

The following shows how to use Workflow Event.

8.1 Concept overview

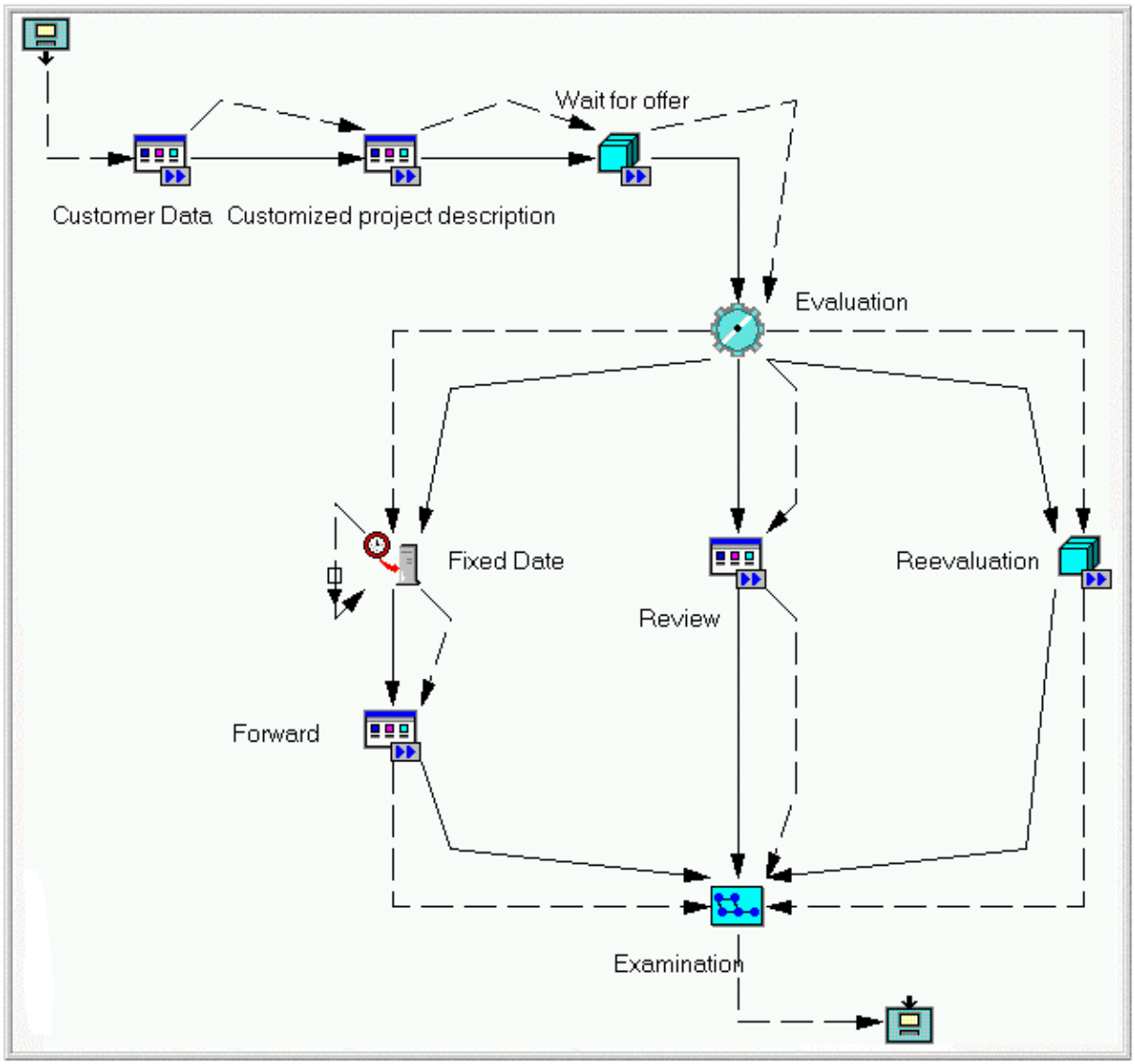


Figure 24: Concept Overview

This example describes a part of a bidding process. In the bidding process a contract is finalized. Sub processes of a bidding process are, for example creation of an offer or the bidding. The creation of an offer contains several sub processes that can be accomplished more effectively with fixed procedures.

The creation of an offer always follows the same procedure (process of creating an offer).

The first step of creating an offer is to add the task description to the system and edit it for company internal processing. After the technical break down of the data, the offer is forwarded for evaluation.

The processing contains several steps. While the File Event the directory can be put under surveillance, the directory can be checked for the arrival of the files. This ensures that the data is complete before the process is continued. The File Event checks for the fixed attributes, the file extensions, file name and dates.

After the arrival of the necessary information, it has to be decided how the offer needs to be processed further. If an offer is complete or it requires the offer further editing and searching for sub contractors. The decision is based on the project's complexity and priority. Projects of low priority don't have to be reviewed at once, so it is forwarded on a set date. With the Time Event a specific date can be set after which the process is continued. An offer of high priority has to be reviewed at once. If the offer is not complete, another sub process (Block) is started, and the offer is edited till all requirements are met.

Now the offer has to be forwarded at once.

The last step before the bidding the offer has to be reviewed. The review may consist of several steps. But these steps are very specific and depend on the customer situation. For this reason, this task is displayed simplified. After the execution of the last step, the offer can be presented to the customer. This completes the sub process creation an offer.

8.2 Description of the Simulation

8.2.1 Requirements

- Installation of the software described in the chapter 2 "Installation" (see page 5)
- Put the included program `actimpl.exe` in `<C:\Program Files\MQSeries Workflow\BIN>`
- The example FDL `ausschreibung.fdl`

8.2.2 Execution

To display the process model described in this FDL file it is possible to import the *ausschreibung.fdl* into Buildtime. It is important that the user defined Icons (MQWFESF und MQWFEST) have to be known in the Buildtime database (see *"Importing of Icons"* page 13). After importing it is possible to make changes to the simulation.

The first step is to import the process in the Runtime component, the following function call has to be executed:

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

After importing the Runtime database, the user can start the Client and start the model. It is recommended to start Workflow Event Server as Time as well as File Event with in the FDL designed users and let them run in the background.

8.2.3 Starting the example

For this example a specific data structure was designed. For this case, all important proprieties are designed and transported through this data structure.

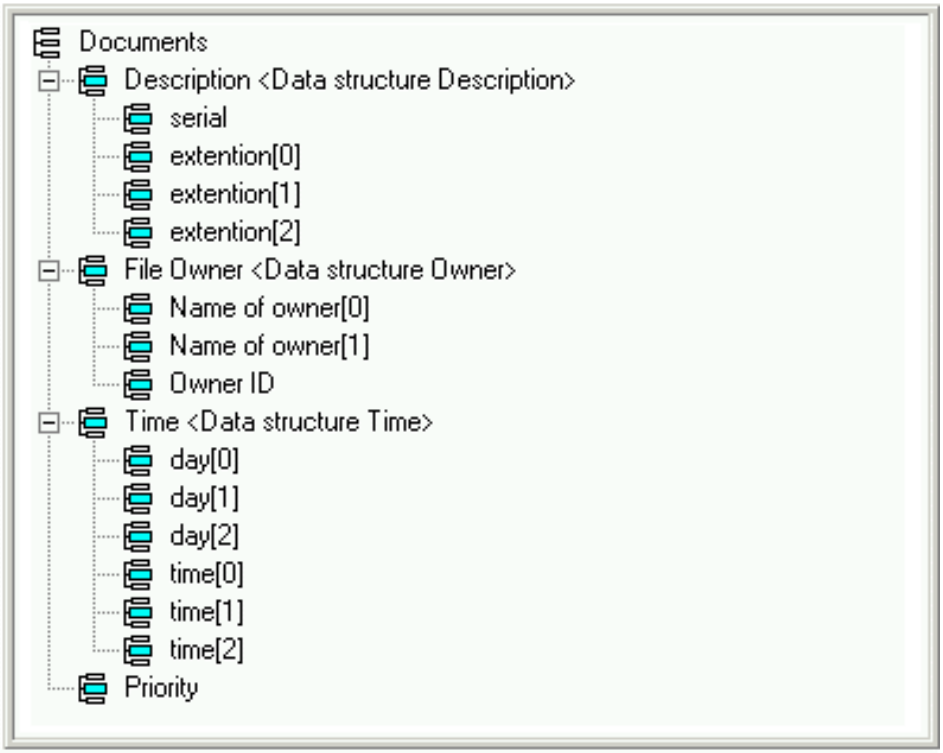


Figure 25: Example data structure

The important properties of a document can be, for example, project-specific serial number, extensions for different documents, the description of the owner, timestamp for processing and priority. After Starting the process, the user is asked to enter the values for the data structure required information.

The data is requested in a dialog window (see *"Figure 26: Data request in the Client* page 35), that displays the information name, data type and a field to enter the value. In the left window part, the single parts (Member) of the data structure can be selected directly.

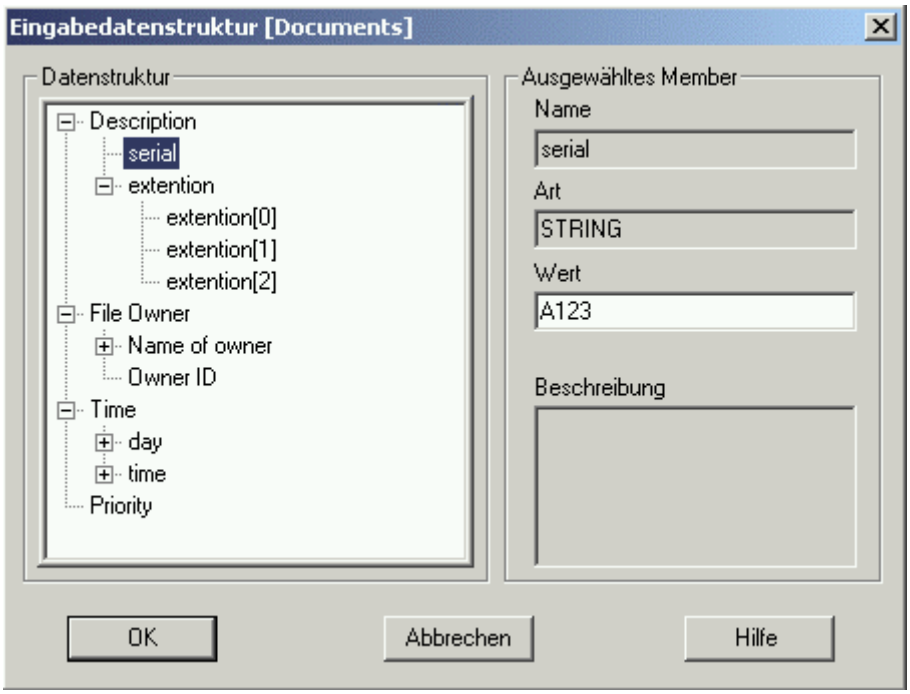


Figure 26: Data request in the Client
This shown data structure is an example.

8.2.3.1 Execution Description

After the data is entered in the data structure and the worklist refreshed, the first activity Customer Data appears on the current worklist. The first step is to start this activity. This activity represents the collection of data of the client and the task. In this example a program is started, that displays the data structure with the contents and requests entering the information (see *"Figure 27: Example program to change the data"* page 35).

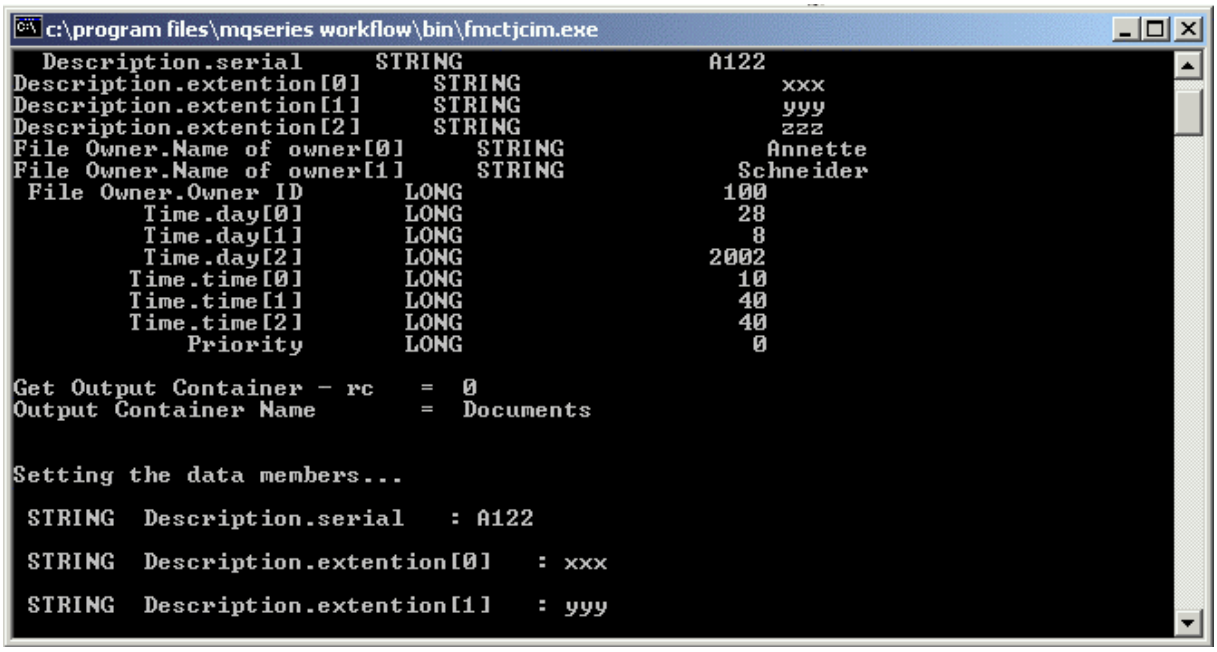


Figure 27: Example program to change the data

The second activity, Customized project description, represents the task of rewriting the data for company internal use. The same application is used like with Customer Data. This time it is important to check the data for completeness and correctness, because in the next step the extension and the serial number are used for the File Event.

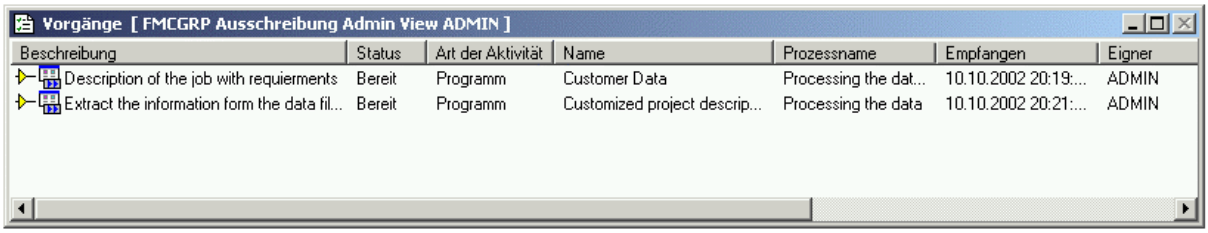


Figure 28: View of Activities in the Client

The next step is the File Event. In this step the process is on hold until all files with the necessary information have arrived. This example assumes that the complete description is organized thematically in three files. The project description is divided in technical information, customer information and in project description. *Wait for offer* combines these three activities in one Block (see *"*

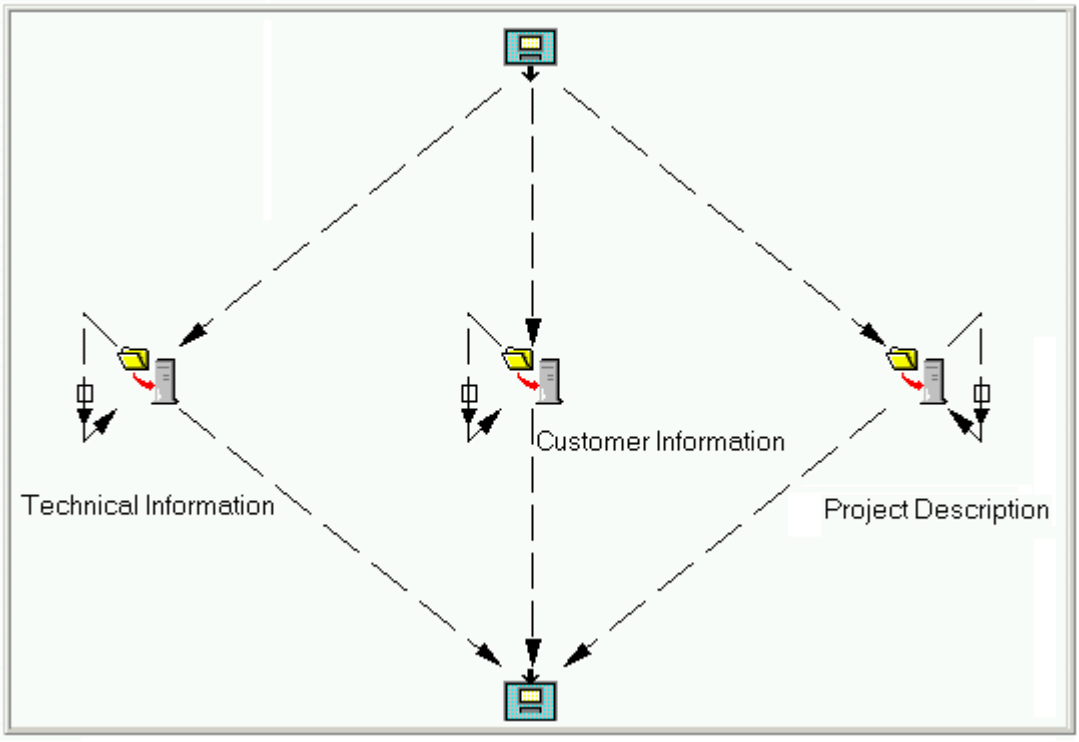


Figure 29: Block Wait for offer*"* page 36).

Figure 29: Block Wait for offer

Each of these activities waits for a file. Only if all three are complete the process will be continued. To make this functionality available, it is necessary that File Event Server is active and the status for the block Wait for offer is "ready" (see *"Figure 30: Activities wait for offer"* page 37). The directory path is <C:\Documents\>. The file names have to contain the data structure described extension and serial. As soon as all documents were found in the directory Wait for offer will be completed.

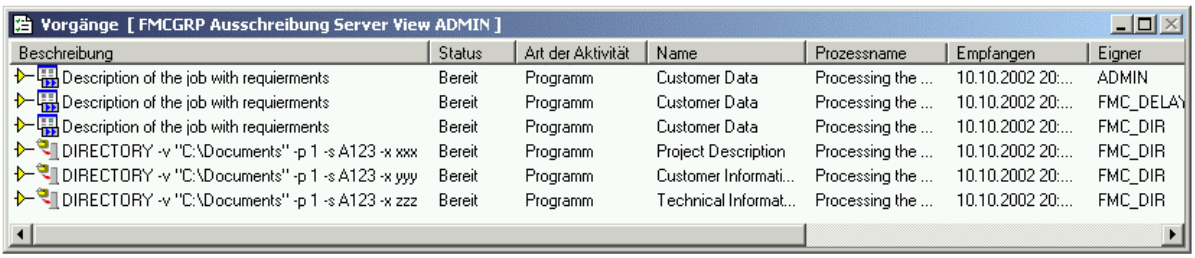


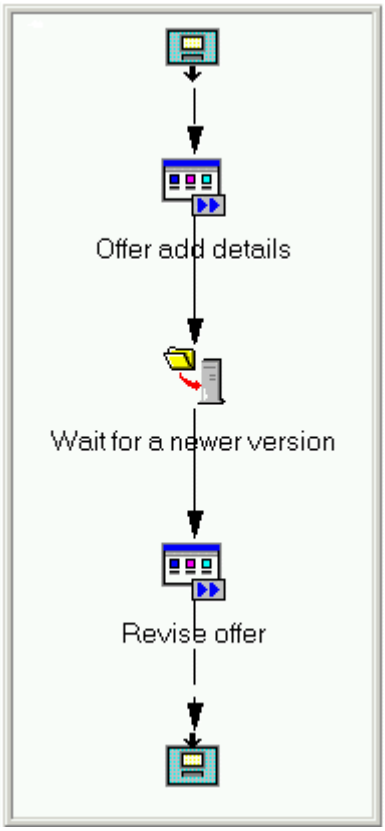
Figure 30: Activities wait for offer

Now the decision process after the priority can begin. During a bidding it is important to set priorities. The process flow of a project is different depending on its importance and complexity. Projects of a lower priority can be processed and reviewed at a later date. Projects of high priority have to be processed at once and projects of high complexity have to be processed further. In this example, the decision is read from the data container of the activity Evaluation.

The Time Event Server is used in the case of the lowest priority. The timestamp, when the process is to be continued, is taken from the data structure. The Timestamp requires the complete date and the exact time including the seconds (see *"Figure 31: Activity Fixed Date"* page 37).

Figure 31: Activity Fixed Date

The process will be continued as soon the set time is reached. Now the offer is forwarded and checked. If the second case is chosen the offer will be forwarded at once and in the third case the offer still has to be edited until all aspects of the offer are completed and it only needs a last check.



1. The offer edit further and missing details added.

2. The File Event continues the process after a newer version of the document was sent to the target directory.

3. This activity checks if the offer is completed. If the offer is completed the main process is continued, if this is not the case, the offer has to be edited further.

Figure 32: Block Reevaluation

After the review, the offer will be turned in.

9 Appendix

9.1 Required Software

This is the required software for a Windows 2000 or AIX system:

- **Programs:** IBM MQSeries Workflow Version 3.3.2[4]
  JavaTM 2 SDK, Standard Edition Version 1.4.0
- **Software packet:** mqwfes.jar, fmcojagt.jar, fmcojapi.jar

9.2 Bibliography

This section lists the recommended secondary literature. In the handbooks MQSeries Workflow published by IBM, it is possible to read more about different aspects of functionality.

- IBM MQSeries Workflow: *Concepts and Architecture*, GH12-6285, explains the basic concepts of MQSeries Workflow. It also describes the architecture of MQSeries Workflow and how the components fit together.
- IBM MQSeries Workflow: *MQSeries Workflow for Windows NT for Beginners*, SG24-5848, contains the installation and configuration, Build, deploy and execute a business process, includes Workflow API programming examples.
- IBM MQSeries Workflow: *Getting started with Buildtime*, SH12-6286, describes how to use Buildtime of MQSeries Workflow.
- IBM MQSeries Workflow: *Getting started with Runtime*, SH12-6287, describes how to get started with the Client.
- IBM MQSeries Workflow: *Installation Guide*, SH12-6288, contains information and procedures for installing and customizing MQSeries Workflow.
- IBM MQSeries Workflow: *Administration Guide*, SH12-6289, explains how to administer an MQSeries Workflow system.

[4] See redbooks IBM MQSeries: Installation Guide for requirements of the operation system and middleware.