Client User Manual

DataMóvil

Version 3.1

Satec DataMovil Client User Manual

Copyright Notice

Trademarks

SATEC, S.A.

Avda de Europa 34 A, 28023 Aravaca - Madrid (España) Tel.: (+34) 91 708 90 00

Index

1. Introduction

1.1. What is DataMovil?

DataMovil is a platform for the rapid development and deployment of applications for mobile devices like PDAs. It mainly addresses applications with wireless network connections like GPRS, CDMA, 3G or Wi-Fi (802.11x), though it can be also used to execute applications local to the device.

It is a platform originally designed for PDAs, based on standards and with utilities that facilitate the integration with corporate backoffice systems. Since 3.0 version, DataMovil is also available for MIDP 2.0 devices.

1.2. What is DataMovil composed of?

DataMovil is composed of the following modules:

DataMovil smart client

It is an application executed in the device (PC, UMPC, PDA, mobile phone…). This application is executed based on data which is received from a server or loaded from a local file, in XML format. This XML incorporates all the data relevant and the actions to be executed over the data, by the end-user or automatically by the client itself.

As a result of the execution of an XML received by the client, DataMovil will send an XML to the server or, alternatively, it may be stored locally. The server may answer sending another XML to the client.

The XML handled by the DataMovil smart client is based on the XForms 1.0 W3C Recommendation of October 14th 2003. The small differences between the standard and DataMovil arises from the computing resources limitations of a mobile device where DataMovil will be executed and to provide with some extra functionality in some aspects related to communications with peripheral devices.

DataMovil Server:

The DataMovil server includes several components that provide different services:

Application downloading and version control. This functionality is provided by a servlet that it is invoked by the client in order to discover what new applications or new versions of already downloaded applications are available for the user in the server.

Engine to include the data sent by DataMovil, in a relational database. Through the edition of a template that links specific XML tags to tables and fields of any relational database, the engine may incorporate all the data received by the client.

Additionally, the developers may program their own servlets to send a DataMovil-XForms answer to the client at any time.

Administration and development of DataMovil applications:

DataMovil includes several tools as a help to the development of applications. A web based administration is responsible for the registry of the applications created and the proper maintenance of the different versions of an application, including their current status.

For the application development, the web-based editor tool has been changed to a Java "swing" based interface, executed at the desktop by the developer. This new tool provides the following improvements:

- Based on Java, improves the developer productivity.
- Supports the whole XML included in DataMovil.
- Quick preview of individual elements or a group of elements for standard and mobile devices.
- Includes the new published DataMovil XML Schema.
- Synchronization with the administration in order to import/export applications.

1.3. How does DataMovil work?

In order to understand how DataMovil works, we may think about DataMovil comparing it with the traditional web applications, realizing this way its differences and advantages.

In the first place a web application is based upon a web browser that interprets the HTML language composed of tags focused on the presentation of documents on a computer screen. In addition, the web browser incorporates scripting languages like JavaScript to include some processing logic. The data, the presentation and the processing logic are all mixed up in a web application.

In the case of DataMovil, the client is based on XForms; that is an XML language that has a clear separation between the data model and the processing logic. It does not include either presentation logic or scripting languages. The processing logic is included in the XML but the presentation logic is not incorporated to the XForms standard. In the current version of DataMovil, the presentation is automatically handled by the client though it may be configured through the "data.ini" file. From version 2.1 on, the presentation and layout of the different elements of each application may be customized at will by the developer.

These DataMovil features greatly diminish the development time and the maintenance costs as it eliminates the well known problems associated to the maintenance of scripting code and, largely due to the separation of the data model and the processing logic.

The DataMovil smart client communicates with the server in a similar way to a web browser sending and receiving data. The main difference is that DataMovil always sends data in XML format and receives data in XML according to the DataMovil-XForms language based on XForms. An application may be built by means of a sequence of exchanges with one or several servers, the same way as we build applications based on a web browser.

However, the DataMovil smart client presents a number of differences with a web browser, in addition to the already mentioned, that makes DataMovil more flexible and adapted to the mobile environment. A user as he/she starts DataMovil in the mobile device, has the option to execute already loaded applications (files in DataMovil-XForms format) or to connect to a server and download the applications. The user may save the result data locally in the mobile device or send them to a server. In case the user has data saved in the mobile device, he/she can send them to a server at any time a

connection to the server is available. In this way, the user is protected against possible lack of connectivity at any time. In the same way, a user can save the state of an application locally, in the mobile device, at any time (for example, saving the result of an intermediate state of an application composed by multiple interactions with a server) and to recover that state later on to continue the execution of the application.

Another differential feature of the DataMovil smart client is that the client is able to handle a single DataMovil-XForms in a sequence of device screens, minimizing the number on interactions with the server. On the other hand, this makes more friendly the user experience with the device, reducing the annoying scroll bar manipulations. The DataMovil smart client shows automatically the buttons that allow the navigation through the different screens.

The DataMovil server provides the management of the files created in DataMovil-XForms format that allows a user to list and select the applications available for him and their different versions. This feature in included in a servlet that the client invokes to access to “remote applications”. This servlet returns the updated list of applications. Once the user has chosen one application, it can be stored in the device and, then, executed. If the application has several interactions with the server, these interactions require the development of the corresponding servlets the same way as we would proceed in the case of a web application, taking into account that the servlet must return a DataMovil-XForms.

This is the very summarized functioning of DataMovil, which must be taken into account to implement a mobility solution.

2. The DataMovil Client Software.

The DataMovil Client Software is a Java application that is executed in the PDA. The Client needs a compatible Java Virtual Machine installed in the handheld to be executed. The client is available in two implementations, one for CDC/Personal Profile PDAs and another for MIDP mobile phones.

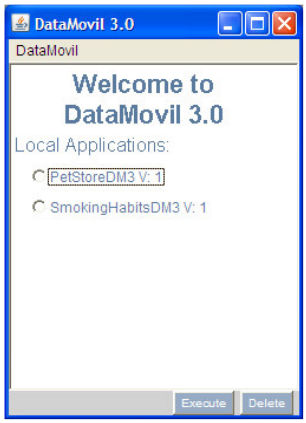
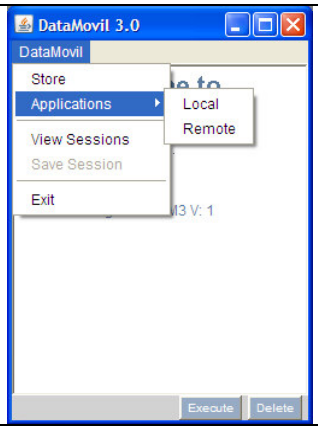
This program is responsible for the processing of DataMovil applications that are DataMovil-XForms files downloaded from the server.

Moreover, the client is responsible for other functionality included in DataMovil:

1. **Download** new applications or new versions of applications from the DataMovil Server.
2. **Maintain and manage** a local storage for the applications.
3. **Maintain and manage** a local storage for data that could not be sent to the server.
4. **Maintain** a local storage for the "**sessions**" of an application which execution is stopped by the user. The user can restart the application session through a menu option.
5. **View and start** an application local to the device.
6. **Maintains local caches** of the execution of an application and its resources (local data files or images).

3. Starting the DataMovil Client.

3.1. PDA

To start the DataMovil client, double click on the DataMovil icon that is on the Init menu of the PDA. This action will launch the JVM and, then, the DataMovil client.	
The first screen includes a title text and a list of the local applications currently downloaded in the PDA. This behavior may be changed to start a default application by modifying the INIT_FORM parameter in the Data.ini file. The title text is set in the MSG_WELCOME parameter of the Data.ini file. At the right low corner of the screen there are two buttons. One is to start the execution of the selected application and the other is to delete it from the local applications storage.	
There is a popup menu, common to all screens of DataMovil, which allows the user to go to: ▪ The local data store. ▪ The local applications store, or to connect to the DataMovil Server to retrieve a list of new applications or new versions of applications to be downloaded by the user. ▪ The local application session store. ▪ The user may also exit DataMovil in this menu.	

3.2. MIDP

The procedure for MIDlet installation may vary depending on the handset model. Please refer to the manual provided by the manufacturer for specific instructions on installing the DataMovil client MIDlet.
After the installation is performed, there will be a *DataMovilME* client MIDlet available in the handset's Java execution environment.

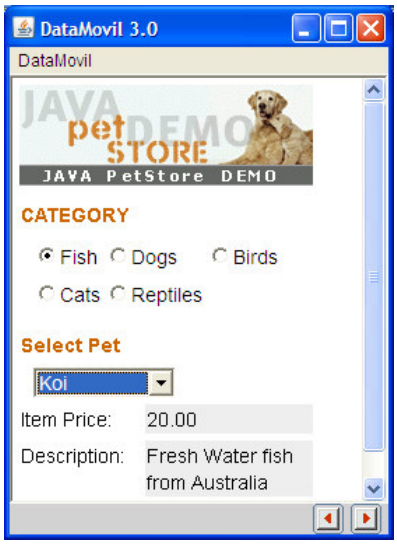
Upon execution, the DataMovilME client shows the **Local Apps** menu. The **s**creen displays the applications currently deployed in the client as a radio button list. The user can run or delete the selected application from the soft-key commands and menu (rendering and order of the options depends on the handset model). Selection is changed activating the control on screen.	
Using the **Remote** command the user can access the server to install more applications. A list of the currently available applications will be shown. Any number of them can be selected and installed using the **Download** command.	

4. Moving through the application screens.

There are two ways to go from one screen to the next in DataMovil. The first one is through the application programming with the “setfocus” action. The second one is through the navigation controls ‘next’ and ‘previous’ available on every screen of an application.

4.1. PDA

In the PDA implementation of the client, navigation is performed with the arrow buttons located at the right low corner of the screen.



The state of the application is maintained as we go forward or back in the applications screens. In case there are no more available screens (pressing ‘previous’ on the first one or ‘next’ on the last) an alert text is shown..

4.2. MIDP

The MIDP client has “Next” and “Prev” commands in every screen of the application. Rendering depends on the handset’s MIDP implementation.

Usually “Next” will be activated directly using a soft button while “Prev” is grouped with the rest of the commands in the menu.

Scroll Mode

In case the application screen is larger than the physical display, a scrolling view mode is available using the **Scroll** command.

In this mode the user can move the view using the cursor keys, without any effect on the application control focus.

To return to the normal view, activate the **Back** command with the soft key (left key in the figure, actual rendering depends on handset model).

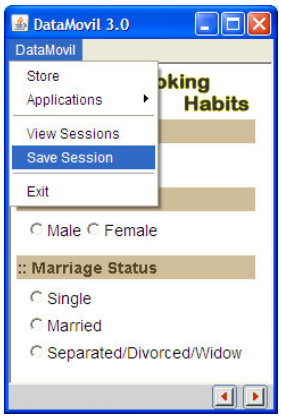
The state of the application is maintained as we go forward or back in the applications screens. In case there are no more available screens (pressing ‘previous’ on the first one or ‘next’ on the last) an alert text is shown..

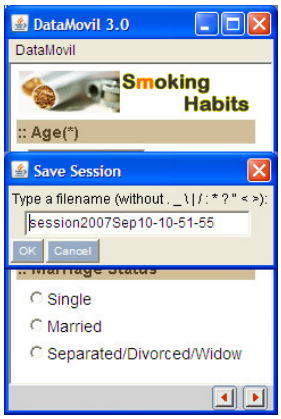
5. The DataMovil "sessions"

The sessions in DataMovil enables the user to save the state of a DataMovil application and exit to the "Local applications" screen.

5.1. Saving sessions.

The user can save the session of a DataMovil application using the menu item "Save application session" while the application is running.



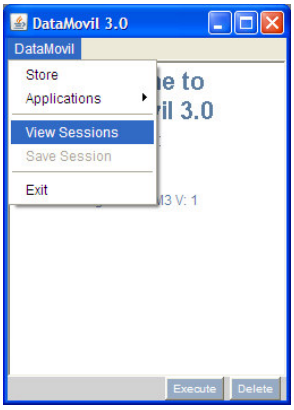


The suggested filename is configured in the SESSION_DEFAULT_NAME variable of the Data.ini file. In the figures above it wasset to 'sess-$APPNAME$-$jnow(HH-mm)$'..

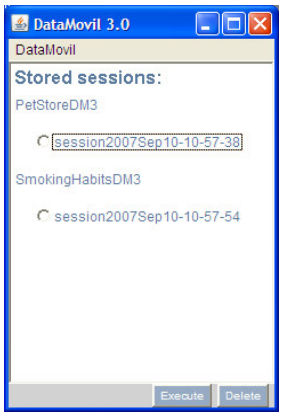
After the session is saved the "Local Application" screen will be displayed.

5.2. Loading and deleting sessions.

The "View application sessions" menu goes to the "Saved Application Session" screen. For PDA this menu is available at any moment in the DataMovil menu.If this option is used while a DataMovil application is running the user will be asked to save a session (see previous section). In MIDP, the menu is accessible from the **Local Apps** screen.



Menu access from main screen.



The "Saved application sessions" screen.

In the "Saved Application Sessions" screen appears all stored sessions for all applications. Besides there are two buttons (or commands, in MIDP) at the bottom-right corner: "Delete" and "Execute".

With the "Delete" button the user can delete the selected session, and with the "Execute" button the user can launch the owner DataMovil application, restoring all saved data.

6. The DataMovil client communications with the server

The DataMovil client communicates with the server under several different circumstances:

1. To download new applications or new versions of applications.
2. To send locally saved data to the server.
3. Through the execution of an XForms "send" action according to the application logical programming.

The behavior of the communications may be controlled by the values of some constants in the Data.ini file.

The **FLAG_VERSION_CONTROL** parameter controls whether all versions of an application are locally saved in the PDA or only the latest. The possible values are "1" to save only the latest version and "0" to save all versions.

The **FLAG_USE_ZIP** parameter controls whether the data communicated between the server and client will be compressed using the ZIP algorithm or not. The possible values are "1" for zipped communication and "0" for uncompressed communications. Unavailable for MIDP.

The **FLAG_AUTOMATIC_DOWNLOAD** parameter controls whether the new applications or new versions of applications are automatically downloaded as the user realizes a submission to the server. The possible values are "1" to automatically download the applications and "0" to suppress the automatic download.

The **FLAG_AUTO_SEND** parameter determines if all data in the local store (it will be explained later) must be automatically sent to the server after a successful data submission occurs. If this flag is set to "1", the data will be sent, and if the flag is set to "0" only the user can send the local store data.

The **FLAG_DATAMOVIL_PROTOCOL** parameter determines whether the DataMovil client will communicate with the DataMovil server using DataMovil proprietary tags or following strictly the Xforms defined protocol. The usage of the DataMovil protocol allows for portal solutions that enforces all the traffic to go through a single access point.

7. The local store management

DataMovil is a system thought to have a communication link between the client and a server. However, sometimes it may not be possible to send data to the server or the user wants to do it at a later time. In order to cover this functionality, the DataMovil client has a local storage area for application result data. This store is used under the following circumstances:

- The application executes a "send" action and the communication link with the server cannot be established for any reason.

In this case the user is prompted to accept or reject to save the data locally. Then, a file name is requested from the user.

This behavior may be changed through the FILE_DEFAULT_NAME parameter of the Data.ini file. If this parameter is left blank, the behavior will be as it was described before. If a value is provided, the DataMovil client will show the name assigned to the data and the user can accept it or overwrite it.

- The FLAG_AUTO_SEND parameter of the Data.ini file is set to "0". In this case the user, whenever a "send" is processed, is prompted to decide whether he wants to save locally the data or to connect to the server and send it. If the parameter is set to "1", the processing continues as aforementioned.

The FILE_DEFAULT_PARAMETER allows the automatic creation of filenames freeing the user of the cumbersome task of typing characters in the PDA. The grammar accepted by this parameter is the following:

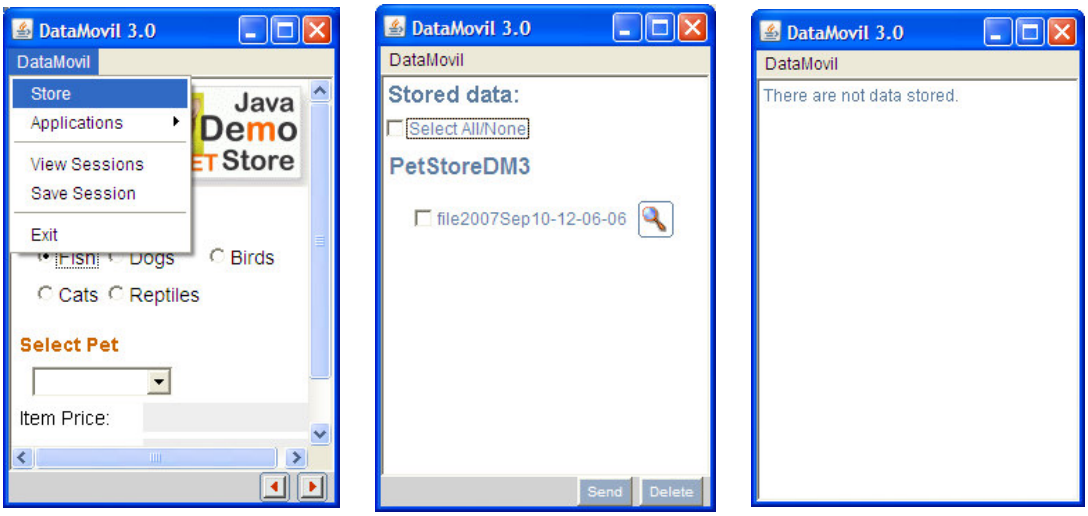
grammar := ($APPNAME$ | $jnow(...)$ | ^[. _ \ | / : * ? " < >])*

The patterns accepted by the $jnow()$ function are described in the Section 4.4.2 of the Programming Manual.

Examples of valid FILE_DEFAULT_PARAMETER are:

- Survey15-$jnow(MM-dd-yy)$
- $APPNAME$-$jnow(hh-mm#dd)$
- $jnow(dd hh-mm)$-PO-SATEC

Once the files with the data saved are created, they can be managed going to the "store" option of the popup menu (see the left image). The next screen shows all saved files in the store. If there are no files , then the right screen is shown.



In this screen we can see the files created for each application and we can even inspect its contents. The following actions can be made on the files:

- Select one, several or all files through the checkboxes.
- View the content of the file through the zoom icon.

- Delete the selected files with the "Delete" button located at the right low corner.
- Send the files to the server through the "Send" button located at the right low corner. If more than one file is selected, they will be sent to the URL defined in the SEND_URL parameter of the Data.ini file. If only one file is selected, it will be sent to the URL defined in the corresponding "submission" element of the application that created the file.

In MIDP the Store management interface looks very similar to the session management one.

The **Pending Store** menu option opens Store management.	Pending submissions are grouped by the application they belong to.	The user can submit or delete a specific item, or send all the stored items at once to the default URL.
		

8. The Data.ini file

The Data.ini file is described in detail in Section 6.3 of the Installation and Configuration Manual. This file serves several purposes:

- Localize the DataMovil client in different languages.
- Adapt the messages shown in DataMovil to different application domains.
- Parametrize different constants used in the system to different specific end user environments

The file is divided into different sections, containing related parameters:

- [SERVER]. It contains parameters related to the communication with the server
- [CLIENT]. it contains parameters related to the directories to be used by DataMovil
- [SIZES]. It contains the sizes of the popup windows or textbox areas
- [COLORS]. It contains the color of the background, texts, buttons, and text buttons
- [FONT]. It contains the face and sizes to be used by default by the applications for button text, title, label components and the options of select components.
- [LAYOUT]. It contains the tab value size, margins for the DataMovil window and the default spacing between components.
- [GLOBAL]. It contains some global constants like the text shown in the upper left corner of the DataMovil execution window, the initial application to be started if any, characteristics of the date formats for input and output date-type data, connection timeout, and default filenames patterns for locally saved data and snapshots
- [MESSAGES]. It contains the text of the different messages that may be shown by the applications. The main purpose of this section is localization.
- [DIALOGUES}. It contains the text in the alert window that may be shown by the applications. The main purpose of this section is localization.
- [FLAGS}. It contains parameters that affect the behavior of DataMovil like whether the communications will be compressed, several versions of an application will be locally saved, new applications will be automatically downloaded or the user will be prompted to save data locally or not.
- [UPLOAD]. It contains data related to the particular configuration of the “upload” component for the capacities of the device
- [MENU]. It contains the text of all DataMovil (not the application) defined buttons and popup menus. Message texts associated to the previous buttons are also included in this section. The main purpose of this section is localization. It also includes offset data to place de forward and back arrow buttons horizontally in the screen. This allows customizing DataMovil for JVMs

that suppress the lower bar of the PDA and only show the virtual keyboard activation icon. This icon may overlap the arrow buttons if the offsets are not properly set.

All these parameters constitute an extensive configuration framework for DataMovil. Some of them have been explained in the previous chapters of this document. The document "Installation and Configuration Manual" contains a detailed description of all parameters. Other parameters of special relevance to the end user are:

- SERVER_URL. Server URL where the DataMovil client connects to download applications and versions of applications. It includes only the host name and port. For example, SERVER_URL=http://goliath.malab.satec.es:83
- VERSIONS_URL. Directory relative to SERVER_URL where the servlet that serves applications is located. For example: VERSIONS_URL = /applications/versionControl
- LOCAL_USER_PATH. Directory of the PDA where all the applications information is stored. For example, LOCAL_USER_PATH = /DataMovil/apps/
- LOCAL_APPL_DIRECTORY. Directory of the PDA relative to LOCAL_USER_PATH where the applications are stored. For example, LOCAL_APPL_DIRECTORY = cuest2
- LOCAL_DATA_DIRECTORY. Directory of the PDA relative to LOCAL_USER_PATH where the application data store is located. For example, LOCAL_DATA_DIRECTORY = encu2
- LOCAL_SAVE_DIRECTORY. Directory of the PDA relative to LOCAL_USER_PATH where the application snapshots are located. For example, LOCAL_SAVE_DIRECTORY =
- LANGUAGE. Language used to process date and time information. Coded according to ISO-639. For example, LANGUAGE = en
- COUNTRY. Country used for the processing of date and time. It is coded according to ISO-3166. For example, COUNTRY = US
- CONNECTION_TIMEOUT. Time in seconds which the DataMovil client waits to connect to the server before releasing the attempts. For example, CONNECTION_TIMEOUT = 40

Incorrect values of these parameters will make that the system will not work.

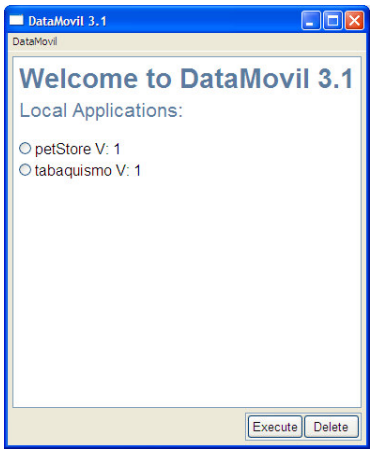
9. The DataMovil Client SWT

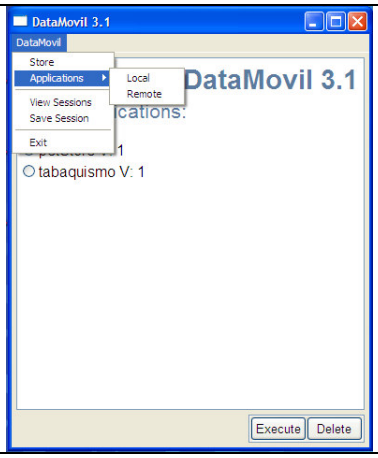
9.1. Installing the DataMovil smart client SWT

The installation in a PDA with the Windows Mobile operating system is realized following these steps:

- Connect the PDA to the PC through the cradle. The ActiveSync must be active.
- Execute DataMovil3.1.exe file, located in the /DataMovilBrowser/PocketPC directory of the DataMovil CD. This will automatically install the DataMovil client in the PDA.
- At the step *setup type*, must select the *IBM J9 SWT* option.
- Edit the data.ini file installed in the PDA. See chapter *8. The Data.ini file*.

9.2. Starting the DataMovil client SWT

To start the DataMovil SWT client, click on the DataMovil icon that is on the programs menu of the PDA. This action will launch the JVM and, then, the DataMovil client.	
The first screen includes a title text and the list of the local applications currently downloaded in the PDA. This behavior may be changed to start a default application by modifying the INIT_FORM parameter in the Data.ini file. The title text is set in the MSG_WELCOME parameter of the Data.ini file. At the right low corner of the screen there are two buttons. One is to start the execution of the selected application and the other is to delete it from the local applications storage.	

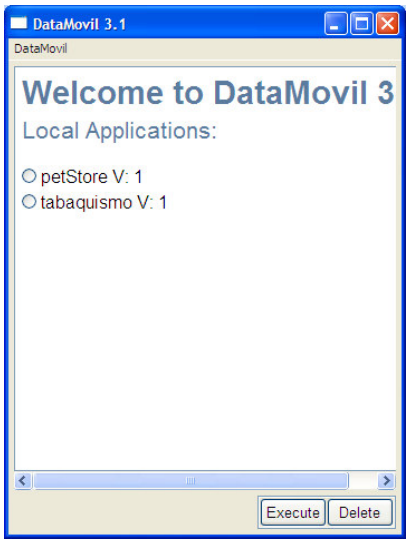
There is a popup menu, common to all screens of DataMovil, which allows the user to go to: ▪ The local data store. ▪ The local applications store, or to connect to the DataMovil Server to retrieve a list of new applications or new versions of applications to be downloaded by the user. ▪ The local application session store. ▪ The user may also exit DataMovil in this menu.	

9.3. Differences between client j9AWT and j9SWT

Welcome screen: In the SWT client the distribution of applications becomes vertical instead horizontal, as in the AWT client. The look of the applications list and buttons are also different.

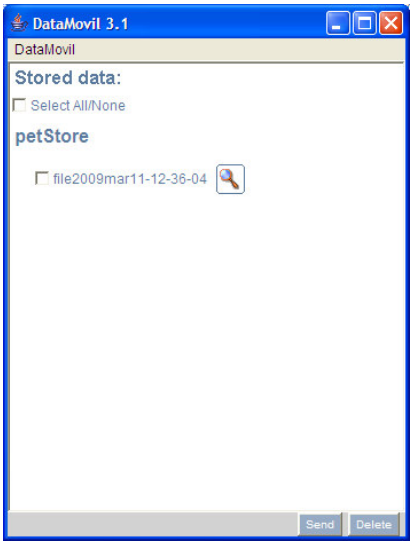
J9AWT

J9SWT

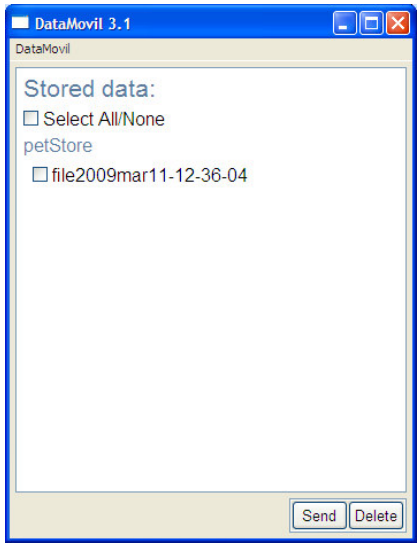


Store screen: changes in text format and styles. The magnifying glass icon to examine data stored in XML disappears.

J9AWT



J9SWT



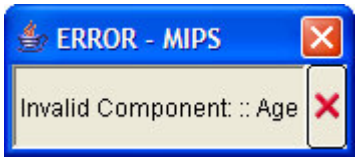
An example of an application screen: changes in text format and styles.
NOTE about the font size: SWT interprets the size of a font in a different manner than the AWT J9/JBed versions. SWT always interprets a size of 12 bigger than AWT interpretation.

J9AWT

J9SWT

Display messages more friendly in swt version.

J9AWT



J9SWT

The difference of font sizes relies on the different point size definition in Windows based presentation (SWT) and in Java based presentation (AWT). Windows systems are based on 96 ppi while Java Virtual Machines use 72 ppi. The result is that SWT fonts sizes are about 1/3 larger than AWT font sizes.