Installation of SAP Mobile Infrastructure 2.5 SP14

ADDON.ME_IG

Release 655

Copyright

Icons in Body Text

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help → General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

Typographic Conventions

Type Style	Description
Example text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example text`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Installation of SAP Mobile Infrastructure 2.5 SP14

Purpose

This installation guide describes how to install the *SAP MI Server Component* on a *SAP Web Application Server* (*SAP Web AS)* and the *SAP MI Client Component* on a mobile device.

Implementation Considerations

Read the SAP Notes about installation before beginning the installation. These SAP Notes contain the most recent information regarding the installation, as well as corrections to the installation documentation.

Make sure that you have the most recent version of each SAP Note. You can find the SAP Notes on the *SAP Service Marketplace* at **`service.sap.com/notes`** or in *SAPNet - R/3 Frontend*.

List of Related SAP Notes

SAP Note Number	Description
852142	SAP Mobile Infrastruct. 2.5 SP14 – composite note
768959	Released MI client/server landscape for SAP ME 2.1/MI 2.5

History of Changes

Make sure you use the **current** version of the Installation Guide.

You can find the current version of the Installation Guide on the *SAP Service Marketplace* at **`service.sap.com/instguides`**.

The following table provides an overview of the most important changes in previous versions.

Version	Important Change
	First version of the SAP Mobile Infrastructure 2.5 SP14 Installation Guide

Integration

The complete *SAP Mobile Infrastructure (SAP MI)* system landscape consists of

- A **mobile device** with an installed **SAP MI Client Component** and **mobile applications**

  The *SAP MI Client Component* includes software developed by the Apache Software Foundation (see Appendix A: Legal Statements of the Third Party Products [Page 130]).

- An SAP MI Server Component that receives data from the mobile device and passes it to the back-end system. It has two parts:
  - The **SAP MI J2EE Server Component** is an integral part of the J2EE stack of the *SAP Web AS.*
  - The **SAP MI ABAP Server Component** is an integral part of the ABAP stack of the *SAP Web AS*.

- **Back-End Systems**

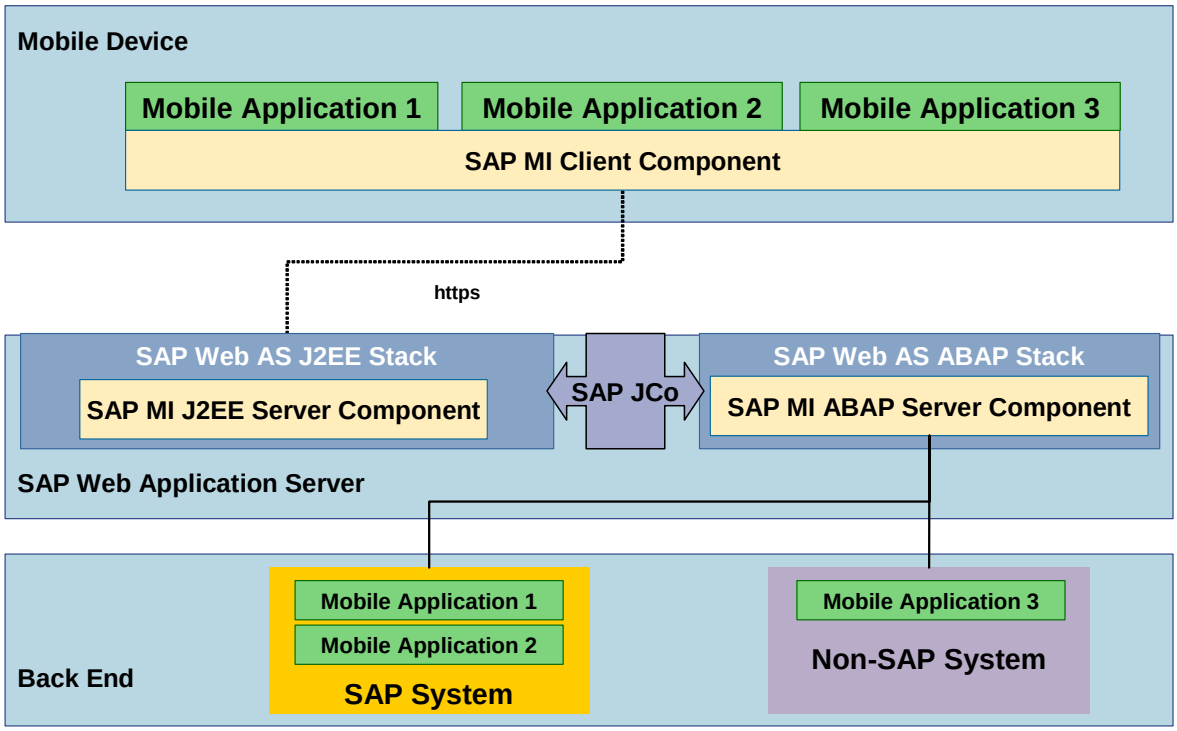
For more information about the system architecture see System Architecture [Page 9].

For reasons of simplicity, the following abbreviations are used in this documentation:

Abbreviation	Full Name
SAP MI	SAP Mobile Infrastructure
SAP MI Client Component	SAP Mobile Infrastructure Client Component
SAP MI Server Component	SAP Mobile Infrastructure Server Component
SAP MI ABAP Server Component	SAP Mobile Infrastructure ABAP Server Component
SAP MI J2EE Server Component	SAP Mobile Infrastructure J2EE Server Component
SAP Web AS	SAP Web Application Server
SAP JCo	SAP Java Connector

System Architecture

The following graphic shows the system architecture of the SAP MI:



SAP MI contains the following technical components:

SAP MI Client Component

The SAP MI Client Component provides a mobile application with the following services:

- UI programming models

  The standard programming model for mobile applications is *Java Server Pages (JSP)*. Alternatively, you can also use the Abstract Window Toolkit (AWT) as a pure Java programming model. The SAP ME 1.0 programming model *microITS* is still supported.

- Framework services

  The framework services are provided to the mobile applications as Java APIs. The most important APIs are used for

  - Data synchronization
  - Data persistence
  - Reading and writing replicated data
  - Logging and tracing
  - Configuration of applications and framework

SAP MI Server Component

The SAP MI Server Component contains the following components:

- **SAP MI J2EE Server Component**

  The SAP MI J2EE Server Component is an integral component of the Java stack of the SAP Web AS.

  - It passes the data containers from the SAP MI Client Component to the SAP MI ABAP Server Component
  - With the SAP MI Web Console, it provides an administration interface to manage the mobile devices and components.

- **SAP MI ABAP Server Component**

  The SAP MI ABAP Server Component is an integral component of the ABAP stack of the SAP Web AS.

  The SAP MI ABAP Server Component is responsible for:

  - Queuing and acknowledgement of synchronized data containers
  - Calling the application logic

    The application logic can be called synchronously or asynchronously, depending on the application.

  - Data replication

    Data replication defines data packages for individual mobile devices (data allocation), determines the data to be newly replicated on the device (delta synchronization), finds and solves conflicts between the mobile device and the server application (conflict management) and provides a number of monitoring tools.

  - Deploying the mobile applications to the mobile devices

    Mobile applications are automatically deployed to a mobile device when the mobile device is synchronized. This process is controlled centrally by the SAP MI Web Console. It permits the system administrator to assign application versions based on users or roles and thus gives an overview of the mobile devices, error logs and so on that are used.

- **SAP Mobile Development Kit (SAP MDK)**

  The SAP Mobile Development Kit (MDK) offers the developer useful documentation and tools for developing mobile applications based on SAP MI. The MDK is part of the *SAP NetWeaver Developer Studio*.

Back End

The back end of a mobile application comprises Customizing and repository objects. Both kinds of objects are transported using the standard mechanisms of the SAP Change & Transport System.

Configuration

Before the end user can work with *SAP Mobile Infrastructure (SAP MI*), you as consultant or system administrator must configure various technical components, upload data from the backend, and edit master data. The following configuration steps are necessary:

- Configuration of SAP Mobile Infrastructure (to be carried out once only)

  After the SAP Web Application Server has been installed, you configure it for use with SAP Mobile Infrastructure and prepare the installation of the mobile devices. You configure the following areas:

  - General Settings [Page 13] (for example, setting up background jobs, role editing)
  - Configuration of Smart Synchronization [Page 24]
  - Configuration of Computing Center Management System [Page 27]
  - Preconfiguration of SAP MI Client Component (Optional) [Page 35]
  - Configuration of Security (Optional) [Page 55]
  - Configuration of Authentication (Optional) [Page 59]
  - Configuration of Framework Deployment [Page 64]

- Configuration of mobile applications (for each mobile application)

  You must carry out additional configuration steps for each mobile application. You can find general information on these configuration steps under Configuration of Mobile Applications [Page 69]. For detailed information, see the installation instructions for the respective mobile application.

- Configuration of mobile devices using device configuration (for groups of users)

  You can configure mobile devices and drivers installed here using device configuration. You can use device configuration for both the initial configuration after the mobile device has been synchronized for the first time and for any subsequent changes. For more information, see Configuration of Mobile Devices with Device Configuration [Page 101].

- Installation of the mobile device using installation images (for groups of users or individual users)

  To speed up the installation of the mobile device you can create setup packages. For more information, see Mobile Device Installation from the Installation Image (Optional) [Page 107]

- Installation of SAP MI on the mobile device (for each mobile device)

  Either you or the user of the mobile device install the SAP MI Client Component on each mobile device using one of the described methods. After the installation has been completed, the device has to be synchronized twice to download the mobile applications and the application data for the user. For more information, see Setting Up SAP MI on the Mobile Device [Page 114].

Prerequisites

Required Landscape Components

You can find the latest information about the required components in SAP Note 670643 at **service.sap.com/notes**.

- **SAP Web AS 6.40** (including the SAP J2EE Engine and the ABAP stack)

  We recommend that you use the latest version of the SAP Web AS 6.40. You can find the most recent versions on the *SAP Service Marketplace* at: **service.sap.com/patches**.

- **Back-end system(s)**
  - If you want to use **Smart Synchronization**: Any SAP System, regardless of its release, and any SOAP-enabled third-party system. Multiple back ends are possible.
  - If you want to use **Generic Synchronization**: SAP Systems based on the releases 4.5, 4.6, 6.10, 6.20, 6.30, 6.40. Multiple back ends are possible.

    If your SAP System is based on the Releases 4.5 or 4.6 you also have to install the **PI 2002.1** or **PI 2003.1** SAP R/3 plug-in. An application on the back-end system that processes the data coming from the mobile device.

Information:

- You can find more information about installing the *SAP Web AS* on the *SAP Service Marketplace* at: **service.sap.com/instguides** → *SAP Web Application Server*.

  A list of the installation guides that are available for each operating system is displayed.
- For information about the platforms supported for *SAP Web AS*, see the *SAP Service Marketplace* at **service.sap.com/platforms**.

Requirements for the SAP MI Client Component / SAP MI Server Component

General Prerequisites:

See *SAP Service Marketplace* at: **service.sap.com/pam** → *SAP NETWEAVER* → *SAP NETWEAVER 04* .

Prerequisites for Using the SAP MI Client Component on a PDA

To use the SAP MI Client on a PDA you need the Java Virtual Machine CrEme, see Uploading Framework Files [Page 67]. Follow the procedure described in Note 772334 to download with an unlimited license.

Prerequisites for Using the Installation Toolkit:

- Operating system: Windows 2000, Windows XP or Windows 2003 Server
- Database: MaxDB 7.5.0.25 For more information about installation and configuration, see SAP Note 765433.

The installation toolkit supports only the JSP version, and not the AWT version.

Prerequisite for Using the Client Installer:

A Win32 operating system or PocketPC operating system is used on the mobile device.

Prerequisite for Installing on Compact Flash Card:

DB2e is used as the database.

Configuration of the SAP Mobile Infrastructure

Use

The following sections contain information about configuring the *SAP Mobile Infrastructure*.

General Settings

Use

The following sections contain information about defining general settings, such as scheduling background jobs and creating user groups.

Reports for Scheduling Background Jobs

Use

You can schedule the following reports on the SAP MI ABAP Server Component using background jobs:

Asynchronous Processing of Inbound Data Containers:

Report Name	WAF_MW_MAPPING
Meaning	The *SAP MI* offers synchronous and asynchronous processing of data containers coming from the mobile device to the SAP MI ABAP Server Component. The type of processing depends on the type of data container defined by the application developer. This report processes the data containers that were defined for asynchronous processing.
Recommended Frequency of Repetition	Daily or more frequently to ensure that data coming from a mobile device is processed as soon as possible. In test phases you should increase the frequency of repetition, for example to every two minutes.

Synchronization of Role Contents with the Contents of the Deployment Console:

Report Name	WAF_DEPLOYMENT_FROM_ROLES
Meaning	The report extracts the applications assigned to the users of a role (in Transaction *PFCG*). The report then transfers the extracted applications to the tables used by the SAP MI Web Console. If you stored RFC destinations in Table *MEMAPPDEST*, the report also evaluates the roles edited in the associated backends. In this case the applications that were assigned to a user in the backend are transferred to the tables used by the SAP MI Web Console. If you created a user group with the name *MESYNC* (see Creating a User Group for Synchronization [Page 15]), the users determined in the backend are automatically generated on the *SAP Web AS* and assigned to this user group. These users initially do not have any authorizations and must be assigned authorizations. This is done by assigning a suitable role (see Role Editing for Mobile Applications [Page 16]).
Recommended Frequency of Repetition	Daily or depending on how often the contents of the roles are changed. In test phases you should start the report manually after a role modification or for example schedule it for hourly.

Activities

1. Schedule the above reports in background jobs (see Scheduling Background Jobs [Page 14]).

   Start Transaction *SE38* to execute reports manually.

   To schedule background jobs you need authorization for the backend. Otherwise the background jobs cannot be executed correctly. Make sure that you have all the necessary authorizations. For more information, see the corresponding R/3 documentation.

Scheduling Background Jobs

Use

You can define and schedule background jobs in two ways from the *Job Overview*:

- Directly from Transaction SM36. This is best for users already familiar with background job scheduling.
- The Job Scheduling Wizard [External]. This is best for users unfamiliar with SAP background job scheduling. To use the Job Wizard, start from Transaction SM36, and either select *Goto* → *Wizard version* or simply use the Job Wizard button.

Procedure

1. Call Transaction SM36 or choose *CCMS* → *Jobs* → *Definition*.
2. Assign a job name. Decide on a name for the job you are defining and enter it in the *Job Name* field.
3. Set the job's priority [External], or "Job Class":
   - High priority: Class A
   - Medium priority: Class B
   - Low priority: Class C
4. In the *Target server* field, indicate whether to use system load balancing.
   - For the system to use system load balancing to automatically select the most efficient application server to use at the moment, leave this field empty.
   - To use a particular application server to run the job, enter a specific target server.
5. If spool requests generated by this job are to be sent to someone as email, specify the email address. Choose the *Spool list recipient* button.
6. Define when the job is to start by choosing Start Condition [External] and completing the appropriate selections. If the job is to repeat, or be periodic, check the box at the bottom of this screen.
7. Define the job's steps [External] by choosing *Step*, then specify the ABAP program, external command [External], or external program [External] to be used for each step.
8. Save the fully defined job to submit it to the background processing system.
9. When you need to modify, reschedule, or otherwise manipulate a job after you've scheduled it the first time, you'll manage jobs from the Job Overview. [External]

   Note: Release the job [External] so that it can run. No job, even those scheduled for immediate processing, can run without first being released.

   For a simple job scheduling procedure, see the Getting Started Guide [External].

 Creating a User Group for Synchronization

Use

If mobile applications are assigned with a role in a backend system, the role synchronization (WAF_DEPLOYMENT_FROM_ROLES) creates a user with the same name without authorizations and with an initial password for each user with this role that does not yet exist on the SAP Web AS. This is only true for the backend systems that exist in table MEMAPPDEST as an RFC connection; see Copying the Contents of Table BWAFMAPP [Page 72].

You must create user group **`MESYNC`** so that you can later find this user and assign it the required authorizations. All automatically created users are assigned to this group. If the group does not exist, the users are not created.

Procedure

1. Start Transaction *SUGR*.
2. In the *User Group* field, enter **`MESYNC`** and choose *Create User Group*.

3. Enter a description of the user group in the *Text* field.
4. Choose *Save*.

Result

All automatically created users are assigned to group **MESYNC**. You can find these users with Transaction *SU10* and assign them the required authorizations together.

Role Editing for Mobile Applications

Purpose

With this process you can assign users authorizations for *SAP MI*. Authorizations are assigned in the *SAP MI* according to the SAP authorization concept. For more information see Users and Roles (BC-SEC-USR) [External] and SAP Authorization Concept [External].

Process Flow

1. You create roles with the required mobile applications or enhance existing roles to include the mobile applications.
2. You assign the following authorization objects to the created roles (see Creating a Single Role [Page 18]):

Authorization Objects

Authorization Object	Field	Value	Description
S_ME_SYNC	ACTVT (Action)	38 (Execute)	Execution of synchronization, relevant for all users
S_RFC	ACTVT (Action)	16 (Execute)	RFC access to all function groups
	RFC_NAME (RFC object to be protected)	RFC1 SDIFRUNTIME SYST SG00 SRFC SYSU	For all individual users (Java Connector)
		ME_USER	For all individual users (To change the synchronization password from the SAP MI Client)
		SUSO	For service users, see Creating Service Users [Page 20] (Determining the relevant error message)

Authorization Object	Field	Value	Description
		MEREP_INSTTK_MPC	For administrators working with the Installation Toolkit, see Mobile Device Installation from the Installation Image (Optional) [Page 107]
		BWAF_MOMO	For administrators (use of the SAP MI Web Console)
		ME_CENTRAL_TRACING	For administrators (tracing in the SAP MI Web Console)
		BWAF_INSTALLATION	For administrators (installation data in the SAP MI WebConsole)
		BWAF_MW	For all individual users (synchronization)
	RFC_TYP (type of RFC object)	FUGR (function group)	
S_TCODE	TCD (transaction code)	SMOMO	For administrators (display data in the SAP MI Web Console)
		MEREP*	For administrators (transactions for Smart Synchronization)
		MI_MCD MCD	For administrators and developers (display and edit Mobile Component Descriptors)
S_MI_MGMT	ACTVT (Action)	* (for all values) 01 (Add, Create) 02 (Change) 03 (Display) 06 (Delete) 78 (Assign)	For administrators (device administration and configuration)
	MI_GROUP	Stored in table MEMGMT_AUTH_GRP, transaction MGMT_AUTHORITY For the definition of groups with different authorizations, e.g. ADMIN and SUPPORT	

Authorization Object	Field	Value	Description
S_MI_CCMS	ACTVT (Action)	*	For administrators (customizing of alerts and display of alerts in the Alert Monitor)
S_MI_ALERT	ACTVT (Action)	36	For administrators (customizing of alerts)
S_DATASET	ACTVT	34 (Write)	For administrators (storage of alerts on the server)

3. You configure user-specific data filtering for the applications contained in the role by assigning the authorizations that control user-specific data filtering [Page 89] to the above roles. The documentation for the applications contains information about the authorizations that must be assigned here.
4. You assign users the corresponding roles. You can combine users into user groups.

Creating Single Roles

Use

If none of the standard roles delivered by SAP meet your needs, and they cannot be adjusted to do so, create your own single role with the following procedure.

Procedure

1. To start role maintenance, either choose *Create Role* in the SAP Easy Access transaction die or *Tools* → *Administration* → *User Maintenance*→ *Role Administration*→ *Roles* (transaction PFCG).
2. Enter the name of the role.

   Roles delivered by SAP start with the prefix "SAP_". For your own user roles, instead of using the SAP namespace, use the customer namespace. This means that the prefix is "Y_" or "Z_".

   You cannot tell from the names of the delivered roles whether they are single or composite roles. You should therefore create a naming convention for your roles so that you can differentiate between single and composite roles.
3. Choose *Create*.
4. Create a more detailed description of the role including, for example, the activities contained within it. You can create role documentation that can be displayed with HTML in the Knowledge Warehouse, and then assign it to the role by choosing *Utilities* → *Info Object* → *Create assignment.* The user can then call the documentation by choosing *Show Documentation* or *Documentation for the role*.

   You can use an existing role as a reference to extend the authorizations of the user. For more information, see Derive roles [External].

5. You can assign transactions, reports, and Web addresses to the role on the *Menu* tab page. The system automatically creates the authorizations that you can set on the *Authorizations* tab page from the transactions that you store in the menu structure of the role. For more information, see Creating a Role Menu [External].

   So that you can call the transactions in another system in a role, enter the RFC destination of the other system in the *Target system* field. If the *Target system* field is empty, the transactions are called in the system in which the user is logged on.

   You should only use RFC destinations which were created using the Trusted System concept (Trusted System: Trust Relationships between SAP Systems [External]) to guarantee that the same user is used in the target system. This is only necessary, however, if you want to navigate using the Easy Access Menu in the SAP GUI.

   If you use mySAP Workplace in the Web browser, you can use any destination containing a logical system with the same name.

   You can also specify a variable which points to an RFC destination. Variables are assigned to the RFC destinations in the transaction SM30_SSM_RFC.

   To distribute the role into a particular target system, specify the target system (its must have a release status of at least SAP R/3 4.6C) and choose *Distribute*.

6. To generate the profile for the role [External], choose *Change Authorization Data* on the *Authorizations* tab page.

   An input window may appear, depending on which activities you selected You are prompted to enter the organizational levels. Organizational levels are authorization fields which occur in a lot of authorizations (an organizational level is, for example, a company code). If you enter a particular value in the dialog box, die authorization fields of the role are maintained automatically.

   The authorizations which are proposed automatically for the selected activities of the role are displayed in the following screen. Some authorization have default values.

   Wherever traffic lights appear in the tree display, you must adjust the authorization values manually. You can maintain the authorization values by expanding the object classes and clicking on the white fields to the right of the authorization field name.

   When you have maintained the values, the authorizations count as manually modified and are not overwritten when you copy more activities into the role and edit the authorizations again. You can assign the complete authorization (*) for the hierarchy level for all non-maintained fields by clicking on the traffic lights.

   Wherever there are red traffic lights, there are organizational levels with no values. You can enter and change organizational levels with *Org. levels*.

   If you want other functions in the tree display, such as copying or collecting authorizations, you can show them with *Utilities* → *Settings*.

   a. Generate an authorization profile for the authorizations. To do this, Choose *Generate*.

      You are prompted for an authorization profile name. A valid name in the customer namespace is proposed.

   b. Leave the tree display after the profile generation.

      If you change the menu and then call the tree display for the authorizations again, the authorizations of the new activities are mixed with those for the

existing authorizations. There may then be a few yellow traffic lights, because there are authorizations in the tree that are incompletely defined. You must either manually assign values to these, or if you do not want to do this, delete them. To delete an authorization, deactivate it first and then delete it.

You can add general authorizations, such as spool display or print with authorization templates to the existing data. Choose *Edit* → *Insert authorizations* → *From template.* Choose a template (SAP_USER_B – Basis authorization for application users or SAP_PRINT – print authorization). You can also create a separate role for clarity.

7. You can also assign users [External] to the role immediately.
8. Save your entries.

Result

You have created a role. A user menu is displayed to the user to whom this role is assigned when he or she logs on to the system. The user has the authorizations which you specified to perform the activities in the user menu.

See also:

- Editing Predefined Authorizations [External]
- Assigning Users a Standard Role [External]
- Changing a Standard Role [External]
- Creating a Composite Role [External]

Creating Service Users

Use

You need the service user for example to display detailed messages in case of logon errors.

Procedure

Defining the Service User Role

Define a role for the service user; see Role Processing for Mobile Applications [Page 16] and Creating Single Roles [Page 18].

1. Create an authorization profile without a template and add authorization object ***S_RFC*** to it.
2. Choose Action 16 (*Execute*).
3. Enter the following as the RFC objects to be protected:
   - **`RFC1`** (for the SAP Java Connector)
   - **`SDIFRUNTIME`** (for the SAP Java Connector)
   - **`SG00`** (for the SAP Java Connector)
   - **`SRFC`** (for the SAP Java Connector)
   - **`SYST`** (for the SAP Java Connector)
   - **`SYSU`** (for the SAP Java Connector)

- **SUSO** (Function group for determining the detailed error message)

4. Select *Function Group* as the type of the RFC object to be protected.
5. Generate the profile and save the role you created.

Defining the Service User

1. Start Transaction *SU01* and create a technical service user (e.g. MI_SERVICE).

   For the password, only use the characters contained in the ISO 8859-1 character set.

2. Assign the role you created to the service user.

Creating Property Files

Use

With this procedure you can provide the MI Sync Servlet with the necessary connection data for the SAP MI ABAP Server Component.

Prerequisites

As administrator, you must perform these steps on the computer on which the *J2EE Engine* is installed. You also need a user with authorization to administer the J2EE Engine and the corresponding password.

Creating Property Files

1. Start the J2EE Engine.
2. Start the J2EE Visual Administrator and connect to the J2EE Engine. To do this you need the administrator password.
3. Choose the service *Configuration Adapter* from the *Server* → *Services* menu.

   Choose the *Runtime* tab page, followed by the *Display Configuration* tab page.
4. Navigate in the tree to *Configurations* → *apps* → *sap.com* → *com.sapmarkets.mesyncjco* → *appcfg*
5. Change into editing mode by confirming the popup with *Yes.*
6. Click *appcfg* with the secondary mouse button and choose *Create Subnode.*

   You must provide a property file for each SAP Web AS (SAP MI ABAP server component) you want to connect to. This file must contain the application server and system number of the corresponding SAP Web AS.
7. Make the following

8. entries:

Entries in the *Create* dialog box

Field	Entry
Dropdown Box	Select *File Entry*
Name	**<sysid>.props**, for example, `abc.props` The name of the property file must be identical to the system ID of the SAP Web AS to which you want to connect. For example, if the system ID of the *SAP Web AS* is `m25`, the name of the file must be `m25.props`. Enter the name of the file in lowercase letters.
Display	Select *Text* in the dropdown box and specify the content of the file, for example: **ashost=myserver** **sysnr=06**

   Alternatively you can upload a file that already exists. Choose *Upload* to do this. The name and the contents of the file you uploaded appear in the dialog box.

9. Choose *Create* to generate the properties file.
10. Change into the display mode and leave the J2EE Visual Administrator.

You have created the property file and can now test the synchronization.

Testing Synchronization

To test if synchronization is working without having to install the client (SAP MI Client Component), call the following URL in your browser:

http://<Server>:<Port>/meSync/servlet/meSync?~sysid=<System ID>&~login=<User name>&~password=<Password>&~client=<Client>&~language=<Language>&~acknowledge=X&~test=true&

http://p59298:50000/meSync/servlet/meSync?~sysid=u6b&~login=tester&~password=test&~client=800&~language=de&~acknowledge=X&~test=true&

If the test was successful, the browser displays a message containing a time that, for example, looks like:

```
&WAF_SYNC&STATUS=&Execution Time =103129& HeaderType =
MEREPLICATION &MORE_PACKAGES_WAITING = &
```

If the test fails, the system generates an error message.

Alternatively, you can test synchronization using a configured client.

See also:

For more information on the Visual Administrator, see Visual Administrator [External].

Configuring Session Handling

Use

To prevent exceptions from occurring because of excessive storage requirements on the SAP MI J2EE Server Component, you can configure session handling. You can define the following parameters:

- Maximum length of a session (timeout)
- How many sessions are possible simultaneously, that is how many mobile devices can be synchronized at the same time

Procedure

Defining the Timeout

1. Open the file *web.xml* in the directory .../*misyncservlet/web-inf* or, if it is not available, in the directory *apps**sap.com**com.sapmarkets.mesyncjco**servlet_jsp**meSync**root**WEB-INF*.
2. Enter a value in minutes under `<session-timeout>`. The value should roughly correspond to the length of a synchronization cycle. The default is 30 minutes.
3. Save your entries.

Defining the Maximum Number of Sessions

1. Open the file *web.xml* in the directory .../*misyncservlet/web-inf* or, if it is not available, in the directory *apps**sap.com**com.sapmarkets.mesyncjco**servlet_jsp**meSync**root**WEB-INF*.
2. Insert the <max-session> parameter into the file:

```
</session-timeout>
</session-config>
<max-sessions>
MAXIMALE ANZAHL
</max-sessions>
</web-app>
```

3. Replace **`MAXIMALE ANZAHL`** with the required maximum number of sessions.
4. Save your entries.

Displaying Detailed Messages for Logon Errors

Use

You can configure the SAP MI J2EE Server Component so that the cause of the error can be determined and a detailed error message output on the SAP MI Client Component if server logon errors occur.

Prerequisites

You created a service user, see Creating Service Users [Page 20].

Procedure

1. Open the system file created under Creating Property Files [Page 21].
2. Make the following entries:

   ```
   MobileInfrastructure.Sync.ServiceUserName = <Service User>
   ```

   ```
   MobileInfrastructure.Sync.ServiceUserPassword = <Password of Service User>
   ```

3. Save your entries.

   If you change the password at a later time, you must adjust the file again: Delete the entry **`MobileInfrastructure.Sync.EncodedServiceUserPassword = <encrypted password>`** and add the following entry again:

   ```
   MobileInfrastructure.Sync.ServiceUserPassword = <password of service user>
   ```

Result

The configuration is completed. The password entered in legible text is replaced with an encrypted password with the next synchronization.

If an error occurs when you log onto the SAP Web AS, a detailed error message is now displayed.

 Configuration of Smart Synchronization

Use

The following sections contain information about configuring Smart Synchronization.

 Configuring Receiver Control Records

Use

With the receiver control record you can influence the behavior of the receiver at runtime. The receiver control record must be completely configured before the mobile device can be synchronized with the SAP MI ABAP Server Component.

Procedure

1. Start Transaction *merep_pd.*
2. Choose the *Runtime Component* tab.
3. Choose  *Display <-> Change*.
4. Under *Receiver*, select *Enabled*.
5. Set *Log Level* to **2**.
6. Choose  *Save*.

Configuring Handler Control Records

Use

With the handler control record you can influence the behavior of the handler execution. Since handler processes are normally triggered very frequently and each process is relatively time-consuming, it is important that you set the right value in this handler control record to use the system resources optimally.

Procedure

1. Start Transaction *merep_pd.*
2. Choose the *Runtime Component* tab.
3. Choose *Display <-> Change*.
4. Under *Handler*, select *Enabled*.
5. Set *Log Level* to **2**.
6. Set *Max.Number Handlers* to **5**.
7. Deselect *Loopback.*
8. Deselect *Batch user may differ from logon user*.

   When you schedule a handler background job, you must set this flag. If the flag is not set, the handler prevents the background job from being executed.

   The scheduled handler job selects the data packages of the relevant user on the mobile device or in the backend, but edits them with the user name that the job scheduled with Transaction *SE36*. The authorization checks also refer to the batch user.
9. Choose *Save*.

Scheduling Background Jobs for the Handler

Use

Since it is possible to limit the number of simultaneous handler processes in order to control system performance, some messages from the client device are not processed immediately when being received by the server (see Setting the Maximum Number of Handlers [External]).

If you schedule a background job for the Handler, the messages that were not processed immediately due to the above mentioned reason can be processed with minimum delay.

Prerequisites

- To activate background processing, mark *Batch user may differ from logon user* when Configuring Handler Control Records [Page 25].

  Only set this flag if you are sure that the batch user may edit the handler worklist items and execute the update BAPI wrappers or the read BAPI wrappers in the backend system.

If the batch job executes the messages, the initiation user is the user who sets the batch job. Thus, this user needs appropriate authorizations in both the *SAP Web AS* and in the backend system where the application data resides.

- You created variants with which background jobs can be scheduled.

Procedure

1. Start Transaction *SM36*.
2. Enter a name for the job under *Job name*.
3. Select **A** for the *Job class*.
4. Specify when the job should begin by choosing *Start condition*.
5. Choose *Immediate*.
6. Select the *Periodic Job* option and choose *Periodic value* to define how often it is repeated.

   You have to define a frequency for the handler background job, taking the *Max.Number Handlers* setting of the handler control record into account. If the value is small, set the handler job so that it is not executed frequently, e.g. every half hour.
7. Select the corresponding period value and choose  *Save* to save the period and to return to the *Start Time* screen.
8. Choose  *Save* in the lower part of the *Start Time* screen to return to the *Define Background Job* screen.
9. Choose *Step* and then *ABAP program*.
10. In the *ABAP Program* section, enter **MEREP_RETRY_BP** in the name field.
11. Enter a suitable defined variant, e.g **P_HNDLER**.
12. Choose  *Save* in the lower part of the *Create Step 1* screen to return to the *Define Background Job* screen.
13. Choose  *Save* to save the completely defined job and pass it to the background processing system.

    Release the job so that it can be executed. No job, not even one that is scheduled for immediate processing, can be executed if it was not released.

For more information see Scheduling Background Jobs [Page 14] and Releasing Jobs [External].

 Configuring Sender Control Records

Use

With the sender control record you can influence the runtime behavior of the sender process.

Procedure

1. Start Transaction *merep_pd.*
2. Choose the *Runtime Component* tab.

3. Choose *Display <-> Change*.
4. Under *Sender*, select *Enabled*.
5. Set *Log Level* to **2**.
6. Set *Maximum Bundle Size* to **50**.
7. Choose *Save*.

Configuration of the Computing Center Management System

Use

With the Computing Center Management System (CCMS), the administrator can monitor the systems of the *SAP MI* with Transaction *RZ20* or the path *Tools* → CCMS → *Control/Monitoring* → *Alert Monitor* under *SAP Mobile Infrastructure Templates* → *Mobile Infrastructure*.

Features

The following information can be displayed for the *SAP Mobile Infrastructure* (*SAP MI*):

- Configuration: Configuration data for devices on which the SAP MI Client Component, SAP MI Server Component and backend systems are installed. This includes for example information about the processors and storage capacities.
- Heartbeats: Availability data of the backend systems, SAP MI Client Component, and SAP MI J2EE Server Component.
- Job Monitoring: Information about monitored background jobs
- Alerts: Messages of the SAP MI Client Component These are for example warnings if a certain percentage of the remaining storage is exhausted.
- Tracing: Log information of the SAP MI Client Component and SAP MI Server Component. This includes for example error messages. You can also delete error messages from the tables and enter threshold values for warnings in this area.

Functions offered by the SAP MI that support work with the CCMS:

- **Alert Monitor** (Transaction *MI_ALMON*):

  With the alert monitor you can display the alerts that were reported to the CCMS. You can make your selections according to various categories. The entire history is displayed.

- **Reorganization Tool** (Transaction *MI_ALBACK*):

  You can clean up alert files occasionally with the reorganization tool. You can select and deactivate alerts according to various criteria or store them at a given location. The stored alert files can be imported again as needed.

Activities

You must perform the following configuration steps to monitor the *SAP MI* with the CCMS.

Configuration Steps

Configuration Step	Description
Schedule job SAP_CCMS_MONI_BATCH_DP.	Transaction *RZ21, Technical infrastructure* → *Method Execution* → *Activate background dispatching*

Configuration Step	Description
Set Tracing for the monitored system. The end user can make this setting for the SAP MI Client. Alternatively, the administrator can set the tracing. The data can only be transmitted to the server if the end user sent the trace to the server and synchronized it.	For information about administrator settings, see Predefining and Setting Parameters for All Users [Page 36]. For information about end user settings and sending to the server, see Configuring the Trace for Error Correction [External].
Define which background jobs should be monitored by the CCMS.	Setting Background Job Monitoring [Page 28]
Configure monitoring of the SAP MI J2EE Server Component	Setting MI J2EE Server Component Monitoring [Page 30]
Adjust the alerts to your needs.	Customizing for Alerts [Page 34]
Assign authorizations for reorganizing alerts.	Role Processing for Mobile Applications [Page 16]
(Optional) Adjust the frequency of execution of job SAP_CCMS_MONI_BATCH_DP.	Transaction *SM37* or *Tools* → *CCMS* → *Jobs* → *SM37 Maintenance* The value should be between 30 and 60 minutes. The default value is 60 minutes.
(Optional) Change the properties in the Alert Monitor, e.g. for the tracing table *Client table counter*.	Transaction *RZ20* Select the node and choose *Properties*. For more information, see the documentation about the CCMS under Alert Monitor [External].

The storage requirements in shared memory of the CCMS increase with the number of clients and implemented alerts. If needed, you can change the storage settings for your profile with Transaction *RZ10* under *Extended maintenance*. See also Note 135503.

Configuring Monitoring of Background Jobs

Use

If you want to monitor background jobs using the Computing Center Management System (CCMS), you must configure the CCMS appropriately.

Prerequisites

You configured background jobs. See also Reports for Scheduling Background Jobs [Page 13].

Procedure

1. Start Transaction *SE16*.
2. Enter table name **ALBTCMON**.
3. Choose  *Display table contents*.
4. Choose  *Execute*.

5. Choose *Table Entry* → *Create*.
6. Enter the name of the background job you want to monitor.
7. Enter **`MI_JobMonitoring`** as MTE class.
8. Save your entries.
9. Repeat steps 5 to 7 for each additional job you want to monitor.
10. Activate job monitoring; see Configuring Job Monitoring with the Alert Monitor [Page 29].

Result

The CCMS monitors the job and displays the status under *Job Monitoring*.

Setting Up the Monitoring of Jobs with the Alert Monitor

Use

To be able to use the Monitoring of Jobs with the Alert Monitor [External], you must perform the following steps:

- Specify the jobs that are to be monitored
- Activate the data collection method that collects the data of the selected jobs.

These two steps are described here. We also recommend that you create a separate monitor to allow a clear display of the data and maintenance of the threshold values.

Procedure

Select the jobs that are to be monitored

The table ALBTCMON contains name patterns of the jobs that are to be monitored using job monitoring. To monitor the desired jobs, enter the relevant name patterns in the table:

1. Start transaction SE16.
2. The *Data Browser: Initial Screen* screen appears. In the *Table Name* field, enter **`ALBTCMON`** and choose *Table Contents (Enter)* ().
3. The *Data Browser: Table ALBCTMON* selection screen appears. Choose *Execute* (). To create a new name pattern, choose the *Create* button ().
4. The system displays the *Insert Table ALBTCMON* screen. In the *JOBNAME* input field, enter the desired name pattern. In the simplest case, a name pattern can be the name of the job that you want to monitor. You can use the wild card character (*****) to select multiple job names. Leave the other fields empty; they are reserved for future developments.
5. Save your changes.

For each name pattern, the system creates a subtree, in which it displays information about the status and runtime of the corresponding jobs.

Monitoring jobs with the monitoring architecture is always system-local. If you have a central monitoring system, you must nevertheless always make the selection of the jobs to be monitored in the local system.

Activating the Job Monitoring

By default, Job Monitoring is deactivated. To activate it, you must ensure that the corresponding data collection method starts to collect data automatically. To do this, follow the procedure below:

1. From the SAP Easy Access Menu, choose *Tools* → *CCMS* → *Configuration* → *Alert Monitor*, or call transaction RZ21.
2. The *Monitoring: Properties and Methods* screen appears. In the *Methods* group box, select *Method Definitions* and choose *Display Overview*.
3. The system displays an overview of the method definitions. Select the method *CCMS_BATCH_MONITORING* and choose *Edit Data* ( ).
4. The *Monitoring: Methods* screen appears. Choose *Display* ↔ *Change* ( ). Choose the *Control* tab page, and in the *Startup Method* group box, activate the *Execute Method Immediately After Start of a Monitoring Segment* indicator.
5. Save your changes.

The data collection method will become active at the next restart of the system and will automatically generate the relevant subtrees.

If you require the job monitoring data immediately, you should reset to WARMUP status [External] the monitoring segment of the central server of your system (the server with the Enqueue service).

Result

The desired data is collected and stored in a monitoring context. This means that it is, for example, visible in the *All Monitoring Contexts* monitor of the *SAP CCMS Technical Expert Monitors* monitor set. However, we recommend that you create a separate monitor to display the data and adjust the alert generation to your own needs there using threshold values and filters (see Background Job Monitoring Monitor [External]).

Configuring Monitoring of the SAP MI J2EE Server Component

Use

The Computing Center Management System (CCMS) monitors the SAP MI J2EE Server Component using Generic Request and Message Generator (GRMG) technology. You must perform the following steps in order to display availability information for the SAP MI J2EE Server Component under *Heartbeat Monitoring* → *SAP MI Server Component* → *SAP MI J2EE Server Component*:

- Adjust GRM customizing
- Upload the template for GRMG customizing into the central monitoring system
- Start the GRMG scenario

Procedure

Adjust and Upload GRMG Customizing

1. Proceed as described in the procedure of The GRMG Runtime Control in the Visual Administrator [Page 31].

Select the following GRMG customizing in the GRMG runtime control: *com.sapmarkets.mesyncjco*.

2. Adjust the following information:

Parameters to be Adjusted in GRMG Customizing

Field	Input
URL/Destination	**`http://<Host>:<port>/meSync/servlet/com.sap.ip.mi.misync.ccms.GRMGApplication`**  The host and port are entered automatically when the scenario is uploaded. You need not enter them.
R3System	Name of the R/3 system (SAP MI ABAP Server Component)
R3Client	Name of the R/3 client (SAP MI ABAP Server Component)
Username	User to be used for the logon to the R/3 system. The user needs synchronization authorization ***S_ME_SYNC*** and authorization for RFC ***S_RFC***.
Password	Password for the above user

3. Upload the GRMG customizing (see Saving Your Work. Uploading the XML to the CCMS [Page 33])..

The GRMG Runtime Control in the Visual Administrator

The J2EE Engine Visual Administrator provides a runtime control for modifying GRMG settings. The runtime control consists of two panes: *Applications with GRMG customization* and *Customizing tree for application*.

To access the GRMG runtime control, choose *Dispatcher / Server → Services → Monitoring → GRMG Customizing*.

When you start the Visual Administrator, the system searches the database for stored *grmg-customizing.xml* files and lists the applications that have a deployed GRMG customization in the *Applications with GRMG customization* pane. At this stage, all applications with available GRMGs are displayed. It is not important whether the GRMG is properly written or not.

When an application is selected, an XML tree is built in the *Customizing tree for application* pane. The tree consists of two types of nodes: tag nodes () and text nodes (). If there are missing tags containing text, the system automatically creates them and sets a default value for them. If there are no scenarios, components, and properties tags under the customizing, scenario, or component tags, they are added but no sub-tags are added under them. If there are tags that are not defined in the GRMG DTD, they are ignored.

If one or more tags are reiterated and must appear only once when you select the application, an error message is displayed. You have to delete the repeated tags from the XML file manually and then redeploy the application.

For more information about how you can use the *GRMG Customizing* runtime, see:

Adding Tags [External]

Editing and Removing Tags [Page 32]

Saving Your Work. Uploading the XML to the CCMS [Page 33]

See also:

Enabling the Availability Monitoring of J2EE Engine [External]

Availability Monitoring of J2EE Applications [External]

Displaying the Monitored Data in the CCMS [External]

Editing and Removing Tags

Use

The GRMG *Customizing* runtime gives you the ability to remove scenario, component, and property tags along with their content, and to edit text tags.

Procedure

From the *Applications with GRMG customization* pane, choose the application whose GRMG XML you want to change.

If you want to	Then
Edit a text tag	1. From the *Customizing tree for application* pane, browse the tree and choose the text tag you want to edit. 2. Choose *Edit* from the toolbar. The Edit text dialog appears. 3. In the field, change the value of the selected tag. Choose *OK*.
Remove tags	1. From the *Customizing tree for application* pane, browse the tree and choose either the scenario, component, or property tag. 2. Choose *Remove* from the toolbar.

Saving Your Work. Uploading the XML to the CCMS.

Use

After you have finished with the XML configuration, you can store the file in the database and also upload it to the CCMS agent.

Procedure

Saving Your Work

Select an application and choose the *Save* option – the changes made to the configuration are stored in the database. This option is useful in case you are planning to make further changes to the configuration.

Use *Save* in order to save your changes to the database. Otherwise, when choosing the *Refresh* option, or if the connection to the server is lost, your last changes will be lost.

Uploading the XML to the CCMS

When you have a final configuration that you are not planning to change anymore, choose the *Upload* option on the toolbar to upload the XML file to the CCMS agent.

The upload can be performed only if you have a CCMS agent installed.

In the CCMS agent installation directory there is a folder named *grmg*, where the XML files are uploaded.

At the time of deployment, the XML file is validated according to the GRMG DTD [External]. If the validation passes successfully, the file is stored in the CCMS agent and in the database. If the validation fails, an error message is displayed prompting you to rewrite your tags according to the GRMG specification. The name of the file in which the configuration is stored must start with the string `GRMG_` in order to be noticed by the agent. The directory in which these files are stored is:

- */usr/sap/temp/grmg* – for Unix systems
- */saploc/prfclog/grmg* – for Windows

After the *Upload* option is selected and the GRMG XML is stored in the CCMS Agent, the file must be transported to an R/3 system. Sometimes this transportation takes nearly an hour. This is a result of the fact that the R/3 system sends requests to the CCMS Agent for new *grmg-customizing.xml* files at regular intervals of one hour. Therefore, if you have uploaded your XML immediately after an R/3 request has passed, you will have to wait up to one hour for the transportation.

See also:

Displaying the Monitored Data in the CCMS [External]

Customizing for Alerts

Use

The following alerts are included in the shipment:

- MaxDBFreeLogSpace: Storage for the log information. If there is not sufficient space available, the system outputs a message.
- MaxDBFreeSpace: If a certain part of the given amount of disk space for the SAP DB is exhausted, the system outputs a message.
- MaxDBNumberofBadIndices: If the number of incorrect indexes is exceeded, the system outputs a message.
- TracfileSize: If a certain percentage of the file size defined in the properties file for the trace file or the entire amount is exhausted, a message is output.
- LastSuccesfullSync: Time between the last successful synchronization and the current synchronization. If the synchronization fails, the system outputs a message.

Activities

You can adjust the threshold values for the alerts in Customizing with *SAP NetWeaver* → *SAP Mobile Infrastructure* → *Creating and Configuring Alerts.*

Preconfiguration of the SAP MI Client Component (Optional)

You can, if required, preconfigure the *SAP MI Client Component*. You can replace files, add additional files, and adjust the configuration. You then create a new installation file for the installation on the mobile device.

Preconfiguring on Windows32 Platforms

Copying the File Structure of the SAP MI Client Component

Use

Before you modify the file structures, you must create a copy of the original that can be modified.

Procedure

1. Open the folder on the DVD *SAP NetWeaver '04 - Additional Components for SAP BW, SAP MI, SAP XI, SAP KW / MI*, containing the file structure of the *SAP MI Client Component*. You can find the file structure in directory *client\asw* bzw. *client\jsp*.
2. Copy the entire file structure including sub-folders and their contents, for example to directory *C:\temp* on your hard disk.

Setting the Screen Mode of the SAP MI Client Component

Use

You can define that the SAP MI Client Component should always start in full screen mode, in minimized mode, or as a service in the background by adding files to the installation.

If you only want to start the SAP MI Client Component once in full screen mode or minimized mode, you can do this with the following call:

`mobileengine.exe -f` for full screen mode

`mobileengine.exe -s` for minimized mode

Procedure

1. Create an empty text file named **`fullscreen.txt`** for full screen mode, **`minimizedFile.txt`** for minimized mode, or **`startasservice.txt`** for service mode.
2. Copy the newly created text file into the directory. Add the file to the directory ***\uncomp\program files\SAP Mobile Infrastructure\Ext1***, see the Adding Files and Directories [Page 52].

3. Start the file setup.exe in the directory ***uncomp***.
4. Start the SAP MI Client Component and check if the file starts in the configured mode.

Result

The SAP MI Client Component is always started in full screen mode, minimized mode or as a service.

Predefining and Setting Parameters for All Users

Use

You can predefine or set certain parameters for all users in file `MobileEngine.config` of the copied file structure.

Procedure

To define certain parameters for all users, you must edit file `MobileEngine.config` in directory `<SAP MI installation directory>\settings.` You can do this with a text editor.

After making your adjustments, you must create a new installation file containing file ***MobileEngine.config*** (see Creating New Installation Files [Page 53]). Use this method for the initial installation on the mobile devices.

You can partly change the parameters for mobile devices already being used by means of device configurations (see Configuration of Mobile Devices using Device Configurations [Page 101]). For more information on the various procedures, see Configuration of Mobile Devices [External].

You can change or predefine and add the following parameters:

User Interface of AWT Applications

Parameters	Description
MobileEngine.AWT.EnableDialogboxesResizing	Permit a change in size in the dialog boxes using the mouse or keyboard. Possible values: **`false, true`** Default value: **`false`**
MobileEngine.AWT.EnableScrollbars	Enable scrollbars in dialog boxes. Possible values: **`false, true`** Default value: **`false`**
MobileEngine.AWT.PREFERRED_WIDTH	Define the width of the AWT window. Possible values: positive integers Recommended value: between **`240`** and the maximum screen width Default value: **`240`**

MobileEngine.AWT.PREFERRED_HEIGHT	Define the height of the AWT window. Possible values: positive integers Recommended value: between **290** and the maximum screen height Default value: **290**

Data Compression and Data Packaging

Parameters	Description
MobileEngine.Datacompression.Gzip	Activate and deactivate data compression for synchronization (see Configuring Data Compression [Page 50]) Possible values: **false, true** Default value: **true**
MobileEngine.Packaging.Activated	Activate and deactivate (user-independent) data packaging (see Configuring Data Packaging [Page 49]) Possible values: **false, true** Default value: **false**
MobileEngine.Packaging.VerySmall	Package size for value *very small*. Possible values: positive integers Default value: **20**
MobileEngine.Packaging.Small	Package size for value *small*. Possible values: positive integers Default value: **100**
MobileEngine.Packaging.Normal	Package size for value *normal*. Possible values: positive integers Default value: **2500**
MobileEngine.Packaging.Large	Package size for value *large*. Possible values: positive integers Default value: **50000**
MobileEngine.Packaging.MaxPackageSize	Used package size (user-dependent) Possible values: **0** – **MobileEngine.Packaging.VerySmall** is used **1** – **MobileEngine.Packaging.Small** is used **2** – **MobileEngine.Packaging.Normal** is used **3** – **MobileEngine.Packaging.Large** is used Default value: **2**

User Mode

Parameters	Description
MobileEngine.UM.SingleUserMode	Configure *SAP MI* for single user mode. If you do not set this parameter to **`true`**, you cannot activate the bypass option (**`MobileEngine.Security.BypassLocal LogonPassword=true`**). Possible values: **`false, true`** Default value: **`false`** The setting for this parameter can be changed at runtime. If you change the value from **`false`** to **`true`**, the client checks if more than one user is registered on the device. If multiple users are registered on the client, the client automatically resets the value to **`false`**.

Security and Authentication

Parameters	Description
MobileEngine.Security.HostnameVerifying	Activate or deactivate the check of the host name. If this option is activated, there is an HTTPS connection if the URL host name and the host name specified in the certificate (= *Common Name*-entry of the certificate) are the same (see also Setting Secure Sockets Layer (SSL) Support [Page 55]). Possible values: **`false, true`** Default value: **`true`**
MobileEngine.Security.BypassLocal LogonPassword	Activate or deactivate the logon password query (see Configuring Bypass Option for Logon Password [Page 51]). If this option is activated, the logon password query is deactivated. In this case, the user's system logon is considered to be sufficient authentication. This parameter is not applicable to the Single-Sign-On environment. You can only activate (**`true`**) this option, if **`MobileEngine.Security.Synchronization PasswordHandlingOption=atSync`** or **`once`** and if **`MobileEngine.UM.SingleUserMode=true`**. If these requirements are not met, the client resets the value to **`false`**. Possible values: **`false, true`** Default value: **`false`**

Parameters	Description
MobileEngine.Security.ResetLocal LogonPasswordSupport	Option to reset the logon password (see Configuring Reset Option for Logon Password [Page 51]). If this option is activated, the user can reset his or her password online using the synchronization password. Possible values: **`false, true`** Default value: **`false`**
MobileEngine.Security.Synchronization PasswordHandlingOption	Define handling of the synchronization password. Possible values: **`local`** – Password corresponds to the local password and does not need to be entered for synchronization **`atSync`** – Password does not correspond to the local password and must be entered for each synchronization. **`once`** – Password does not correspond to the local password and must be entered once for each logon. Default value: **`atSync`**
MobileEngine.Security. SyncPasswordFieldAtLogon	Display the field for the synchronization password in the logon screen of the client (SAP MI Client Component). Only possible if **`MobileEngine.Security.Synchronization PasswordHandlingOption=once`** and if **`MobileEngine.Sync.TimedSyncActive=true`**. Possible values: **`false, true`** Default value: **`false`**
MobileEngine.Security.SSLSupport	Activate or deactivate SSL support in the client (see Setting Secure Sockets Layer (SSL) Support [Page 55]). Possible values: **`false, true`** Default value: **`true`**

For information about parameters that are relevant for single sign-on, see Setting Up Single Sign-On on the Mobile Device [Page 59].

General Parameters for Synchronization

Parameters	Description
MobileEngine.Sync.Client	Client in the SAP Web AS Default value: not set

Parameters	Description
MobileEngine.Sync.ConnectionTimeout	Timeout for testing the connection in milliseconds. Possible values: • All positive integers corresponding to a 64-bit number. • **-1** = There is no test for a connection to the gateway. The synchronization data is sent immediately. Default value: **-1**
MobileEngine.Sync.WaitForResultsTimeout	Time in milliseconds that the client waits for a reply from the server (SAP MI Server Component) during synchronization. If no answer arrives from the server during this time, the synchronization process is defined as unsuccessful. Possible values: • All positive numbers corresponding to a 64-bit number. • **-1** = There is no maximum wait time for the server. Synchronization runs until there is a reply. In rare cases, synchronization could run endlessly. Default value: **-1**
MobileEngine.Sync.Country	Current country setting. The changes are valid as of the next time the client is started. Possible values: All country codes corresponding to ISO 3166. Default value: Setting in the operating system
MobileEngine.Sync.Gateway.Protocol	Select the required Internet protocol. The HTTPS protocol ensures secure data transmission. Possible values: **http**, **https** Default value: not set
MobileEngine.Sync.Gateway.Port	Standard port of the SAP J2EE Engine For more information about determining the port, see SAP Note **590956**. Default value: not set
MobileEngine.Sync.Gateway.System	SAP Web AS system name Default value: not set
MobileEngine.Sync.Gateway.Host	Host name of the SAP J2EE Engine Default value: not set
MobileEngine.Sync.Language	Current logon and synchronization language (for example, **EN**, **DE**, **JP**). The changes are valid as of the next time the client is started. Possible values: All language codes corresponding to ISO 639. Default value: Depends on the installation.

Parameters	Description
MobileEngine.Sync.NewProtocol	Use the old synchronization protocol. We recommend that you only change this value if an application only works with the old protocol, because some of the functions of the SAP MI are not supported by the old protocol.  This value can no longer be changed once a synchronization has taken place. Possible values: **`false, true`** Default value: **`true`** (new protocol)
MobileEngine.Sync.Timezone	Current time zone The changes are valid as of the next time the client is started. Default value: not set (system time zone) Possible values: All time zone IDs supported by *java.util.TimeZone*
MobileEngine.Persist .UseSingleByteStringEncoding	Use single byte coding. To be able to work in a Unicode environment, character strings are normally coded using double-byte coding in SAP MI. You can change this setting (see Defining Coding for Synchronization [Page 45])  Only set this value before the client is started for the first time. If you do this later, the data on the mobile device is damaged. Possible values: **`false, true`** Default value: **`false`**

Repetitive Synchronization

Parameters	Description
MobileEngine.Sync. RepetitiveSyncEnabled	Activate or deactivate repetitive data synchronization (see Configuring Repetitive Synchronization [Page 48]). Possible values: **`false, true`** Default value: **`false`**
MobileEngine.Sync. TimeBetweenRepetitiveSyncs	Time between two synchronization calls for repetitive data synchronization in milliseconds, for example, 10 000 for 10 seconds. Possible values: **`0`**, positive integers Default value: **`0`** (= repetitive data synchronization deactivated)

Parameters	Description
MobileEngine.Sync. MaximumNumberOfRepetitiveSyncs	Number of repetitions of synchronization call for repetitive data synchronization Possible values: **0**, positive integers Default value: **0** (= repetitive data synchronization deactivated)

Synchronization By Data Carrier

Parameters	Description
MobileEngine.Sync.ViaFile.Enabled	Activate or deactivate data packaging. The corresponding options are only visible to the use of the client if this option is activated. Possible values: **false, true** Default value: **false**
MobileEngine.Sync.ViaFile.DefaultTargetSizes	Size of the media used for synchronization. The specified values are converted and displayed in the client. Possible values: **<Value in bytes>:Disc;<Value in bytes>:Memory_Stick;<Value in bytes>:<Medium>** If you specify other media, replace spaces with _ (underlining). Do not use any special characters. To display the values in the client, the values are converted: • The byte value is displayed rounded, depending on the value in kilobytes, megabytes or gigabytes. • For **Memory_Stick** and **Disc**, the medium name is converted to the term in the corresponding language. Default value: **1457664: Disc** For example: **MobileEngine.Sync.ViaFile. DefaultTargetSizes = 1457664:Disc;33554432:Memory_Stick**
MobileEngine.Sync.ViaFile.MinimumTargetSize	Minimum medium size Possible values: positive integer starting with **256** Default value: **1,000,000**
MobileEngine.Security.FilterStreams.Encryption	Fully qualified class name of the class for encryption for synchronization using data carriers (see Defining Data Carrier Synchronization [Page 46]).
MobileEngine.Security.FilterStreams.Decryption	Fully qualified class name of the class for decryption for synchronization using data carrier.

Parameters	Description
MobileEngine.Security.FilterStreams.Signing	Fully qualified class name of the class for signing for synchronization using data carrier.
MobileEngine.Security.FilterStreams.Unsigning	Fully qualified class name of the class for designing for synchronization using data carrier.

Time-Controlled Synchronization

Parameters	Description
MobileEngine.Sync.TimedSyncActive	Defines whether synchronization should start automatically at a defined time interval as a background job. If this parameter is **`true`**, you also have to set parameter *MobileEngine.Sync.TimedSyncTimeInterval.* Possible values: **`true`**, **`false`**. Default value: **`false`**
MobileEngine.Sync.TimedSyncTimeInterval	Time between two synchronization calls in milliseconds. Enter for example one minute as **`60000`**. Default value: not set.
MobileEngine.Sync.Background.Filter	Defines if synchronization is only started if there is modified application data. This parameter is only evaluated if **`MobileEngine.Sync.TimedSyncActive=true`** Possible values: **`true`**, **`false`**. Default value: **`false`**

Use of a Proxy During Synchronization

Parameters	Description
MobileEngine.Sync.Proxyhost	URL of the proxy server (only if a proxy is used). Default value: not set
MobileEngine.Sync.Proxyport	Port of the proxy (only if a proxy is used). Default value: not set
MobileEngine.Sync.Proxyused	Defines whether or not a proxy is used. Possible values: **`true`**, **`false`**. Default value: **`false`**

Tracing

Parameters	Description
MobileEngine.Trace.Enabled	Activate or deactivate tracing (user-dependent) Possible values: **`true`**, **`false`**. Default value: **`true`**
MobileEngine.Trace.Filename	Storage location of the trace file (absolute or relative to the *SAP MI* installation file). Possible values: any file name. If the file name begins with `/`, it is interpreted as relative to the installation directory. Default value: `/log/trace.txt`
MobileEngine.Trace.Filesize	Maximum size of the trace file in KB If the file size exceeds the given value, the system creates a backup file with the past traces and starts a new trace file. The old backup file is overwritten. Possible values: • All positive integers corresponding to a 64-bit number. • **`-1`** (no size restriction) Default value: **`200`**
MobileEngine.Trace.Level	Trace level that is currently in effect. Possible values: • **`10`** (System-critical errors only) • **`50`** (Error display) • **`60`** (Warning display) • **`70`** (Information display) • **`80`** (Execution paths display) • **`90`** (Debugging) • **`1000`** (All) Default value: **`50`**
MobileEngine.SAT.Enabled	Activate or deactivate performance trace. If this option is activated, all activities are logged on the mobile device, see Measuring System Performance on the Mobile Device [External]. Since this is very time-consuming, only individual users should make this setting in the client. Possible values: **`true`**, **`false`**. Default value: **`false`**

Language Check

Parameters	Description
MobileEngine.Setting.LanguageCheck	Check if the language of the user is the same as the language of the application. Can be switched off for test purposes. Possible values: **`true`** = Test if the current language of the user is supported by the given application. If it is not supported, the application is not started. **`false`** = Do not test if the current language of the user is supported by the given application. Default value: **`true`**

Loading System Libraries

Parameters	Description
MobileEngine.Startup.LoadDlls.	Load system libraries (DLL) when starting client. The system libraries must reside in the system directory. Use this parameter if system libraries that contain, for example, drivers for printers or scanners should already be loaded at start-up and not only when needed (that is, when something is to be printed). This is particularly relevant to PDAs, as problems can occur here. Possible values: All entries with this prefix are evaluated. The name may not contain endings or path definitions. For example: **`MobileEngine.Startup.LoadDlls.Drucker = prnport, javaapi`** (System libraries *prnport.dll* and *javaapi.dll* in the system directory are loaded when the client is started.)

Defining Coding for Synchronization

Use

To ensure Unicode compatibility, character strings in synchronization are coded using double-byte coding by default. If, however, performance problems occur during Smart Synchronization, you can use single-byte coding. This reduces both the time required to download data (in particular for the initial download) and query data, and the size of the data files.

If single-byte coding is used the system does not perform a validation. Values that are greater than one single byte are truncated.

Prerequisites

- You use Smart Synchronization and file input/output.
- The mobile device is **not** used in a Unicode environment.

Procedure

Only set this value before the client is started for the first time. If you do this later, the data on the mobile device is damaged.

1. Open file ***MobileEngine.config*** in directory <*SAP MI Installation directory*>*settings* with a text editor.
2. Add the following parameter:

   `MobileEngine.Persist.UseSingleByteStringEncoding=true`
3. Save your entries.

Once you have adjusted the file, you must distribute the changes to the mobile devices with a new installation file, see Creating New Installation Files [Page 53].

Defining Data Carrier Synchronization

Use

If the users do not have a stable Internet connection, they can synchronize using data carriers such as memory sticks, diskettes and DVDs. The users store the data on a data carrier and send this data to the administrator. The administrator synchronizes with the back-end systems in the SAP MI Web Console and sends the data carrier back to the users. The users can now update the data on their mobile devices. For more information about the tasks of the administrator, see Synchronizing Data Carriers with the Back End [External]. For more information about the tasks of the user, see Performing Synchronization [Page 127].

So that the users can synchronize using the data carrier, you must configure the mobile devices and the SAP MI Web Console appropriately:

- The mobile device with file `MobileEngine.config`
- The SAP MI Web Console with file `WebConsole.properties`

If you want to use data encryption or data signing, you must also provide the relevant libraries.

- Deploy libraries to the mobile devices as add-ons.
- For the SAP MI Web Console, copy the libraries to directory `<Installation directory of SAP MI>\WEB-INF\lib` or reference it with the J2EE library reference.

The SAP MI uses `java.io.FilterInputStream` and `java.io.FilterOutputStream` for encryption and signing. For information about filter streams, see `http://java.sun.com/docs/books/tutorial/essential/io/filtered.html`.

Implementation class `java.io.FilterOutputStream` is enhanced for signing and encryption. The class needs a public constructor with one argument only (`java.io.OutputStream`).

Implementation class `java.io.FilterInputStream` is enhanced for unsigning and decryption. The class needs a public constructor with one argument only (`java.io.InputStream`).

The SAP MI also offers a factory class for these input and output streams. The factory reads the implementation classes from the MI configuration and creates instances of these classes. If no property is defined, the default implementations are used.

Prerequisites

- At least one application must be installed on the mobile device to allow the end user to synchronize with a data carrier.
- You can only synchronize application data with data carrier synchronization. You **cannot** deploy applications on the mobile device.
- If you want to use encryption and signing, you can use the corresponding third party libraries.

Adjusting the Configuration Files

The following parameters are relevant to data carrier synchronization. For information about the possible values, see Predefining and Setting Parameters for All Users [Page 36].

- For activation of data carrier synchronization, `MobileEngine.Sync.ViaFile.Enabled`
- For the possible media sizes for data carrier synchronization, `MobileEngine.Sync.ViaFile.DefaultTargetSizes`
- For the minimum medium size, `MobileEngine.Sync.ViaFile.MinimumTargetSize`
- For encryption, `MobileEngine.Security.FilterStreams.Encryption`
- For decryption, `MobileEngine.Security.FilterStreams.Decryption`
- For signing, `MobileEngine.Security.FilterStreams.Signing`
- For unsigning, `MobileEngine.Security.FilterStreams.Unsigning`

Adjusting File MobileEngine.config for the Mobile Device

1. Open the file `MobileEngine.config` in the directory `<SAP MI installation directory>\settings` with a text editor.
2. Add the above-specified parameters if they do not yet exist, and adjust them accordingly.
3. Save your entries.

After you have adjusted the file, you must distribute the changes to the mobile device (see Configuration of Mobile Devices [External]).

File WebConsole.properties for the SAP MI Web Console

Data carrier synchronization is already activated for the SAP MI Web Console. You only have to edit the file, therefore, if you want to make further settings.

1. Check if file *`WebConsole.properties`* is in directory `<Installation directory of SAP MI Web Console>`. If it does not exist there, create it.
2. Open the file with a text editor.
3. Add the required parameters (see above) and save your entries.

Deploying Libraries on the Mobile Device

If encryption or signing is required, you must deploy the corresponding libraries to the mobile devices. To do this:

1. Download the libraries of a third-party vendor.
2. If it does not yet exist, create a Mobile Component Descriptor for each library (see Mobile Component Descriptor [Page 84])
3. Upload the library to the SAP MI Web Console (see Uploading Add-On Files [Page 90]).
4. Assign the library to the user or role (see Assigning Mobile Components to Users [Page 99]).

Making Libraries Available to the SAP MI Web Console

Proceed as follows to enable the SAP MI Web Console to access libraries for encryption and signing:

1. Create an archive with the Java classes and name this archive *MISignEncrHook.jar*.
2. Copy the archive to the directory `<Installation directory of the SAP MI Web Console>\WEB-INF\lib`.
3. Open the file *webconsole.properties* from the installation directory of the SAP MI Web Console.
4. Add the following entries:

   For encryption:

   `MobileEngine.Security.FilterStreams.Encryption = com.xyz.MIEncryptionOutputStream`

   `MobileEngine.Security.FilterStreams.Decryption = com.xyz.MIDecryptionInputStream`

   For signing:

   `MobileEngine.Security.FilterStreams.Signing = com.xyz.MISigningOutputStream`

   `MobileEngine.Security.FilterStreams.Unsigning = com.xyz.MIUnsigningInputStream`

   Where `com.xyz.MI*` are implementation classes of a third party vendor of `java.io.FilterInputStream` and `java.io.FilterOutputStream`.
5. Save your entries.
6. Restart the J2EE Engine.

Configuring Repetitive Synchronization

Use

With repetitive synchronization, the data transfer is repeated to completely transfer the data requested by the mobile device to the mobile device. This is especially relevant for data that must first be edited on the server before being transferred.

You as administrator can set the number of repetitions and the interval. The data transfer is continued as defined by the given number of repetitions until all the data is on the mobile device.

The behavior of the application does not change for the user. Depending on the defined interval and number of repetitions, however, synchronization could take a longer time.

As administrator you can define the following:

- Whether repetitive synchronization is used (parameter **`MobileEngine.Sync.RepetitiveSyncEnabled`**)

- The maximum number of data transfers (parameter **`MobileEngine.Sync.MaximumNumberOfRepetitiveSyncs`**)
- The interval in which the data transfer is repeated (parameter **`MobileEngine.Sync.TimeBetweenRepetitiveSyncs`**)

Prerequisites

Repetitive synchronization can only be used in the following cases:

- For applications that are synchronized with Smart Synchronization
- If synchronization is performed using data carriers

Procedure

1. Open the file ***MobileEngine.config*** in the directory ***<SAP MI installation directory>\settings*** with a text editor.
2. Add the above-specified parameters if they do not yet exist, and adjust them accordingly.
3. Save your entries.

After you have adjusted the file, you must distribute the changes to the mobile device (see Configuration of Mobile Devices [External]).

Configuring Data Packaging

Use

If the synchronization link is not stable enough, users of the mobile device can transfer their data in individual packages. If the connection breaks during synchronization and you need to synchronize again, only the packages that did not arrive completely in the target system are transferred.

Users can toggle this option in the *SAP MI Client Component* and select a package size (see Using Data Packaging [External]).

As administrator you can configure data packaging with parameters in file ***mobileengine.config***:

- Activate data packaging (parameter **`MobileEngine.Packaging.Activated`**) This setting is user-independent.
- Define the size of the data packages (parameter **`MobileEngine.Packaging.VerySmall`**, **`MobileEngine.Packaging.Small`**, **`MobileEngine.Packaging.Normal`**, **`MobileEngine.Packaging.Large`**)
- Defined package size (parameter **`MobileEngine.Packaging.MaxPackageSize`**) The setting is saved on a user-dependent basis.

Procedure

1. Open the file ***MobileEngine.config*** in the directory ***<SAP MI installation directory>\settings*** with a text editor.
2. Add the above-specified parameters if they do not yet exist, and adjust them accordingly.
3. Save your entries.

After you have adjusted the file, you must distribute the changes to the mobile device (see Configuration of Mobile Devices [External]).

Configuring Data Compression

Use

Prior to synchronization, the data transferred from the mobile device to the *SAP MI Server Component* and vice versa is automatically compressed. The compression uses the ZIP standard algorithm.

This procedure allows you to switch data compression on or off in the *SAP MI Client Component*. Data compression is active by default.

Procedure

1. Open the file *MobileEngine.config* in the directory <*SAP MI installation directory*>*settings* with a text editor.
2. Add *MobileEngine.Datacompression.Gzip* to the parameters of the file, where applicable.
3. Set parameter *MobileEngine.Datacompression.Gzip* to **`true`** if you want to switch on data compression.

   `MobileEngine.Datacompression.Gzip=true`

   The default value of the *MobileEngine.Datacompression.Gzip* parameter is **`true`**, even if the parameter in the file *MobileEngine.config* was not set.
4. Set parameter *MobileEngine.Datacompression.Gzip* to **`false`** if you want to switch off data compression.

   `MobileEngine.Datacompression.Gzip=false`
5. Save your entries.

After you have adjusted the file, you must distribute the changes to the mobile device (see Configuration of Mobile Devices [External]).

You can check if your settings are correct by switching on a trace prior to synchronization. After synchronization, the trace file contains a message text that tells you if data compression is switched on or off.

`Data compression is switched on`

or

`Data compression is switched off`

If you are using a mobile device with a **Linux** operating system, you can find more information in SAP Note 754221.

Configuring Bypass Option for Logon Password

Use

Where a mobile device is only used by one specific user, it is possible to bypass logon using the logon password. If you activate this option, the user must be authenticated on the operating system. The standard setting for the bypass option is `false`.

This option is not applicable to the single sign-on environment.

Prerequisites

- The device is used in single user mode: `MobileEngine.UM.SingleUserMode=true` (see Predefining and Setting Parameters for All Users [Page 36]).
- The handling option for the synchronization password must not be set to `local`: `MobileEngine.Security.SynchronizationPasswordHandlingOption=atSync` or `MobileEngine.Security.SynchronizationPasswordHandlingOption=once` (see Predefining and Setting Parameters for All Users [Page 36]).

  If the mobile application offers its own control element (pushbutton or link) to start synchronization, this application must support the `atSync` and `once` synchronization options.

Procedure

1. Open the file `MobileEngine.config` in the directory `<SAP MI installation directory>\settings` in an editor.
2. Add the following parameters: **`MobileEngine.Security.BypassLocalLogonPassword=true`**.
3. Save your entries.

After you have adjusted the file, you must distribute the changes to the mobile device (see Configuration of Mobile Devices [External]).

Result

The user does not have to enter his or her local password when calling SAP MI.

Configuring Reset Option for Logon Password

Use

This option allows the user to reset his or her logon password online using their synchronization password. Where this is not possible, the client must be completely reinstalled if the user has forgotten their logon password. The standard setting for the reset option is `false`.

Procedure

1. Open the file `MobileEngine.config` in the directory `<SAP MI installation directory>\settings` in an editor.
2. Add the following parameters:
   `MobileEngine.Security.ResetLocalLogonPasswordSupport=true`.
3. Save your entries.

After you have adjusted the file, you must distribute the changes to the mobile device (see Configuration of Mobile Devices [External]).

Result

The user logon screen contains the pushbutton *Reset Password.* If the user has forgotten his or her logon password, he or she can log on to the server using the synchronization password and enter a new logon password without having to specify the old one.

If the handling option for the synchronization password has been set to `local` (meaning the synchronization password is identical to the logon password), you have to reset the synchronization password first for the user to be able to reset the logon password so that the user can use the synchronization password to verify his or her identity on the server. In this case, the user has to reset the local logon password to the reset synchronization password to keep the two passwords identical.

Adding Files and Directories

Procedure

To enhance the *SAP MI Client Component* with further files and directories or to overwrite existing directories, copy the files and directories into the directory ***\uncomp\program files\SAP Mobile Infrastructure\Ext1***.

After installation, files and directories that you add to this directory are automatically added to *the SAP MI Client Component* installation. If a file is added to a certain directory of the *SAP MI Client Component,* the same directory structure must be set up under ***\Ext1*** as in the original installation. Of importance here is that the directories have identical names.

Files in directories other than ***\uncomp\program files\SAP Mobile Infrastructure\Ext1*** or ***Ext2*** cannot be deleted or added. If this is not the case, the system terminates the installation.

If you defined user-specific default settings for the *SAP MI Client Component*, copy the modified files to directory ***\uncomp\program files\SAP Mobile Infrastructure\Ext2*** (for example, the file *MobileEngine.config*).

The following keys are entered in the Windows registry when you install the *SAP MI Client Component* on Windows32 systems:

- hklm\software\sap\mobile engine\ProductName

- hklm\software\sap\mobile engine\ProductCode
- hklm\software\sap\mobile engine\ProductVersion
- hklm\software\sap\mobile engine\Path

Creating New Installation Files

Use

If you made modifications, you must create a new setup file, which you can upload to the SAP MI Web Console as Win32 framework.

Procedure

1. Pack all modified folders and files into an archive, for example ZIP, ARJ, EXE.
2. Upload the new, packed installation file to the SAP MI Web Console (see Uploading Framework Files [Page 67]).

   You must download ZIP and ARJ files by deploying on the *SAP MI Client Component*. Then unpack the file in a folder and run the setup file.

Preconfiguring on Windows Mobile Platforms

Procedure

1. Select the ZIP file with the CAB files you want to modify. There is one ZIP file for each runtime environment and language. The ZIP files reside in the following directories:
   - JSP runtime environment: *...\Tom\cabfiles\zip*
   - AWT runtime environment: *...\Core\cabfiles\zip*
2. Unpack the zip file that you want to modify.
3. Find directory *BuildDir*:
   a. Open the *inf* file in directory *cabfiles*.
   b. In the file, search for the section `[SourceDiskNames]` and read the directory from there.

      Possible paths are for example:
      `C:\BUILD\GENDIRS\patch_ME\opt\java\packaged\full\Core`

      or `C:\BUILD\GENDIRS\dev\opt\java\packaged\full\Tom`
4. Depending on the runtime environment, create one of the following directories. Use the directory determined in the first step for **`<Directory BuildDir>:`**
   - JSP runtime environment: **C:\BUILD\GENDIRS\<Directory BuildDir> \opt\java\packaged\full\Tom**
   - AWT runtime environment: **C:\BUILD\GENDIRS\<Directory BuildDir>\opt\java\packaged\full\Core**

These directories must be created exactly as described above in order for the CAB file to be created correctly.

5. Copy the extracted directories of the selected zip file to the directory created in the last step or extract the file directly to this directory.
6. Create the following directories in the *TOM* or *JSP* directory. The directories are language-dependent.

   For the JSP version:

   `lang\<language>\settings`, for example, **`lang\en\settings`**

   `startlink`

   For the AWT version:

   `lang\<language>\settings`, for example, **`lang\en\settings`**
7. Copy the following files from the build directory to the new directories:

   For the JSP version:

   - *MobileEngine.config* to directory *lang*\<*language*>*settings*
   - *MobileInfrastructure.url* to directory *startlink*

   For the AWT version:

   - *MobileEngine.config* to directory *lang*\<*language*>*settings*
8. Adjust the CAB files to your needs. You can predefine or set a value for certain parameters (see Predefining and Setting Parameters for All Users [Page 36]).

   To avoid performance problems caused by temporary internet files you can limit their size. To do this you need to change the registry. You can find more information on the necessary settings in Note **851831**.
9. If you do not yet have a Microsoft CabWiz tool, perform the following steps:

   a. Download the Software Development Kit from one of the following Microsoft sites:

      Pocket PC 2002 Software Development Kit

      `http://www.microsoft.com/downloads/details.aspx?FamilyId=2DBEE84A-BD94-4167-B817-2B2E548B2E92&displaylang=en`.

      SDK for Windows Mobile 2003-based Pocket PCs

      `http://www.microsoft.com/downloads/details.aspx?FamilyID=9996b314-0364-4623-9ede-0b5fbb133652&DisplayLang=en`

      Developer Resources for Windows Mobile 2003 Second Edition

      `http://www.microsoft.com/downloads/details.aspx?familyid=6A34DC83-C3CE-4A4C-AB83-491FD5729551&displaylang=en`

   b. Install the Software Development Kit.

      The CabWiz tools reside in the installation directory at: *…\support\ActiveSync\windows ce application installation\cabwiz* or at *…\Windows CE tools\wce420\POCKET PC 2003\Tools*.
10. Copy the following CabWiz tool files to the directory in which you adjusted your CAB files:

    - *cabwiz.ddf*

- *Cabwiz.exe*
- *Makecab.exe*

11. Double-click on file *makecab.bat*.

    This program created a new CAB file in the following directory (depending on the runtime environment).

    - JSP runtime environment:
      C:\BUILD\GENDIRS\dev\opt\java\packaged\full\Tom\cabfiles
    - AWT runtime environment:
      C:\BUILD\GENDIRS\dev\opt\java\packaged\full\Core\cabfiles

12. Upload the new, packed installation file to the SAP MI Web Console (see Uploading Framework Files [Page 67]).

    You must download ZIP and ARJ files by deploying on the *SAP MI Client Component*. Then unpack the file in a folder and run the setup file.

Configuration of Security (Optional)

Setting Secure Sockets Layer (SSL) Support

Use

This procedure allows you to switch SSL support on or off in the *SAP MI Client Component*.

SSL support can also be configured on the mobile device once the *SAP MI Client Component* has been installed.

If you want to install the *SAP MI Client Component* on a mobile device whose operating system is a Windows 32 platform (such as a laptop with Windows 2000), you can execute the configurations described below before distributing the *SAP MI Client Component* by using the uncompressed variant of the *SAP MI Client Component* setup for Windows 32 platforms. If you use the uncompressed variant and want to distribute it using the SAP MI Web Console, you must then perform the steps in Creating New Installation Files [Page 53].

If you do not make any changes, the default values defined below are automatically used.

If you want to install the *SAP MI Client Component* on a mobile device with operating system Pocket PC 2002 or Windows Mobile 2003 (PDAs), you can create the configuration described below before distribution of the *SAP MI Client Component* by modifying the CAB files [Page 53].

If you do not make any changes, the default values defined below are automatically used.

Prerequisites

- You configured the *SAP J2EE Engine* to support SSL. For more information, see **`service.sap.com/webas`** and **`service.sap.com/instguides`** → *SAP Web Application Server*.
- You carried out the steps described in SAP Note **580497**.

Procedure

1. Open file *MobileEngine.config* in directory <*SAP MI Client Component Installation Path*> → *Settings*.
2. Set the appropriate values for the following property keys:

Property Key	MobileEngine.Security.SSLSupport
Possible values	True / false
Purpose	Activate/deactivate SSL support in the *SAP MI Client Component*
Default Value	True

Property Key	MobileEngine.Security.HostnameVerifying
Possible values	True / false
Purpose	Switch the check of the host name on or off. If this property key is switched on, there is an HTTPS connection if the URL host name and the host name specified in the certificate (= *Common Name* entry of the certificate) are the same.
Default Value	True

Property Key	MobileEngine.Sync.ConnectionTimeout
Recommended Value	-1
Purpose	Switch the URL connection test off. This improves the performance when setting up HTTPS connections.

Server Certificates

With server certificate we mean the certificate of the *SAP J2EE Engine* <*Release*> on which the *SAP MI J2EE Server Component* is installed.

The file *truststore*, which is delivered on the DVD *SAP NetWeaver '04 - Additional Components for SAP BW, SAP MI, SAP XI, SAP KW / MI* and is a component of the SAP MI Client Component, contains the root certificate *TC TrustCenter Class 2 CA* of the SAP Trust Center Service [External].

Server Certificates of the SAP Trust Center Service

If you obtained your server certificate from the *SAP Trust Center Service*, no further steps are necessary to make the server certificate trustworthy for the *SAP MI Client Component*. This is

also true for server certificates that you obtained from the certification locations listed in SAP Note 602993.

Server Certificates from a Non-SAP Certification Location

If you obtained your server certificate from a non-SAP certification location that is not listed in SAP Note 602993, you should perform the steps described under Making External Server Certificates Trusted [Page 57].

Deleting Server Certificates

You implicitly confirm your trust in all the certificates contained in the trust store of the SAP MI Client Component. If you do not want to do so for all certificates, you can remove individual certificates from trust store (see Deleting Server Certificates [Page 58]).

Making External Server Certificates Trusted

Use

If you received your server certificate from a non-SAP certification location that is not listed in SAP Note 602993, you must import the server certificate of the non-SAP certification location or its root certificate [External] into the trust store of the *SAP MI Client Component*.

Use Sun Microsystems' Keytool to import the server certificate or root certificate. Keytool is a tool for the administration of keys and certificates.

Prerequisites

- JDK / JRE 1.3.x or JDK / JRE 1.4.x (the Keytool of Sun Microsystems is shipped with JDK / JRE 1.3.x and JDK / JRE 1.4.x).
- Your SAP MI Client Component is installed on a Windows 32 platform or on a PDA with Pocket PC 2002 or Windows Mobile 2003.
- When you install your SAP MI Client Component on a Windows 32 platform, use either the uncompressed variant of the SAP MI Client Component setup to perform the subsequent configurations, or do it on the installed SAP MI Client Component on the mobile device.
- If your SAP MI Client Component is installed on a PDA with Pocket PC 2002 or Windows Mobile 2003, you must already have performed the first steps described under Modifying CAB Files for PDAs [Page 53].

Procedure

1. Copy your server certificate file to <*SAP MI Client Component Installation Path*>*settings*.
2. Start *Command Prompt*.
3. Go to directory <*SAP MI Client Component Installation Path*>*settings*.
4. Enter **`<JAVA_HOME>\bin\keytool -import -alias <alias-name> -file <server-certificate-file> –keystore truststore`**.

You can choose any *alias name*. We recommend that you enter the same name as for the server certificate file.

```
<JAVA_HOME>\bin\keytool -import -alias TestCA -file
TestCA.cer –keystore truststore
```

5. As *keystore password* enter **access** if you are asked to do so in the command prompt.
6. Confirm the *Trust this certificate*? query with **yes**.
7. Delete the server certificate file copied in step 1. This file is no longer needed since its contents were imported into truststore.
8. Check the contents of the trust store by entering the following in the Command Prompt: **<JAVA_HOME>\bin\keytool –list –v –keystore truststore –storepass access**. In this way you can verify that the certificate you just imported exists in the trust store.

Result

The server certificate of the non-SAP certification location or its root certificate is imported into the trust store of the *SAP MI Client Component*.

Make sure that the end user selects the uncompressed variant of the setup, which you configured as described above, when installing the SAP MI Client Component on a Windows 32 platform. The end user must use the modified CAB file in the installation on a PDA with Pocket PC 2002 or Windows Mobile 2003.

Deleting Server Certificates

Use

If you do not trust certificates contained in the trust store of the SAP MI Client Component, you can remove them from the trust store.

Use Sun Microsystems' Keytool to delete a server certificate or a root certificate. Keytool is a tool for the administration of keys and certificates.

Prerequisites

- JDK / JRE 1.3.x or JDK / JRE 1.4.x (the Keytool of Sun Microsystems is shipped with JDK / JRE 1.3.x and JDK / JRE 1.4.x).
- Your SAP MI Client Component is installed on a Windows 32 platform or on a PDA with Pocket PC 2002 or Windows Mobile 2003.
- When you install your SAP MI Client Component on a Windows 32 platform, use either the uncompressed variant of the SAP MI Client Component setup to perform the subsequent configurations, or do it on the installed SAP MI Client Component on the mobile device.
- If your SAP MI Client Component is installed on a PDA with Pocket PC 2002 or Windows Mobile 2003, you must already have performed the first steps described under Modifying CAB Files for PDAs [Page 53].

Procedure

1. Start *Command Prompt*.
2. Go to directory <*SAP MI Client Component Installation Path*>*settings*.

3. Check the contents of the trust store by entering the following in the Command Prompt: **<JAVA_HOME>\bin\keytool –list –v –keystore truststore –storepass access**. Note down the alias names of the certificates you want to remove.
4. Enter **<JAVA_HOME>\bin\keytool -delete -alias <alias name> -keystore truststore**.

   <JAVA_HOME>\bin\keytool -delete -alias TestCA –keystore truststore
5. As *keystore password* enter **access** if you are asked to do so in the command prompt.
6. Check the contents of the trust store by entering the following in the Command Prompt: **<JAVA_HOME>\bin\keytool –list –v –keystore truststore –storepass access**. In this way you can verify that the certificate you just deleted no longer exists in trust store.

Result

The server certificate or root certificate was removed from the trust store of the *SAP MI Client Component*.

Make sure that the end user selects the uncompressed variant of the setup, which you configured as described above, when installing the SAP MI Client Component on a Windows 32 platform. The end user must use the modified CAB file in the installation on a PDA with Pocket PC 2002 or Windows Mobile 2003.

Adjusting the User Settings

Inform the *SAP MI* end users that they should make the following entries under Editing User Settings [Page 126] in field ***Gateway***:

- Enter **https** in place of **http**
- Enter the port on which the *SAP J2EE Engine* for HTTPS is running for **<Port>.** For more information about determining the port see SAP Note **590956**.

Configuration of Authentication (Optional)

Perform the following configuration if you want to support authentication with single sign-on.

Setting Up Single Sign-On on the Mobile Device

Use

With the parameters in the file *MobileEngine.config*, you can configure the client (SAP MI Client Component) to support single sign-on (SSO) if the device has an online connection. The mobile device receives the SAP logon ticket from a system that issues tickets, such as SAP Enterprise Portal. The mobile device can then be verified on the server (SAP MI Server Component) with the SAP logon ticket without the user having to enter an additional password.

If you want to configure the client to use single sign-on, you must define whether the device is to be used by one user (single user mode) or multiple users (multiple user mode).

You can configure the following scenarios:

- One user - SAP MI-oriented

  User authentication is delegated to the ticket-issuing system from SAP MI (AWT and JSP clients)

- One user

  Access to SAP MI using ticket-issuing system, for example, SAP Enterprise Portal (JSP clients only)

- Multiple users

  Access to SAP MI using ticket-issuing system, for example, SAP Enterprise Portal (JSP clients only)

One User - SAP MI-Oriented

The device is used by a single user only. The user starts the client on the mobile device. It requests a ticket that is used for the initial logon and for synchronization from the system that issues tickets. The SAP MI uses the configured URL.

In this scenario users only need to enter a user ID and password when they log onto the system issuing the ticket. The logon data is verified in SAP MI using the SAP logon ticket. Password handling settings are consequently ignored in SAP MI and the user has no access to password management.

In the initial logon, which must be performed online, the user data from the logon ticket is used to create a user in the client.

Parameters and Values for this Scenario

- **JSP Clients:**

  `MobileEngine.UM.SingleUserMode=true`

  `MobileEngine.UM.SAPLogonTicketSupport=true`

  `MobileEngine.UM.SAPLogonTicketBackgroundRequest=true` (optional)

  `MobileEngine.UM.SAPLogonTicketRequestURL=<URL of system issuing ticket>`

  `MobileEngine.UM.SAPLogonTicketWaitingRefresh=<Time in seconds after which the system again checks if the logon ticket was received>`(Default: 3)

  `MobileEngine.UM.SAPLogonTicketRequestTimeout=<Time in seconds between requesting logon ticket and cancellation>` (Default: 90)

  `MobileEngine.UM.SAPLogonTicketLogoffURL=<URL for the logoff>` (optional)

- **AWT Clients:**

  `MobileEngine.UM.SingleUserMode=true`

  `MobileEngine.UM.SAPLogonTicketSupport=true`

  `MobileEngine.UM.SAPLogonTicketRequestURL=<URL of system issuing ticket>`

  `MobileEngine.UM.SAPLogonTicketRequestTimeout=<Time in seconds between requesting logon ticket and cancellation>` (Default: 90)

  `MobileEngine.UM.SAPLogonTicketLogoffURL=<URL for the logoff>` (optional)

  `MobileEngine.UM.ExternalAuthUserParameter`

  `MobileEngine.UM.ExternalAuthPasswordParameter`

  `MobileEngine.UM.ExternalAuthAdditionalParameters`

One User – Access to SAP MI from a Ticket-Issuing System, for Example, SAP Enterprise Portal

This scenario only applies to JSP clients.

The device is used by a single user only. The user starts SAP MI on their mobile devices as a service running in the background without a user interface. There must be an empty file named `startasservice.txt` in the same directory as the file `mobileengine.exe`.

To work with SAP MI, the user opens the SAP MI user interface from a link (for example, in SAP Enterprise Portal).

As a result of logging onto the system issuing tickets, there is already a logon ticket available if the user interface of the SAP MI was started. The logon ticket is, therefore, not explicitly requested.

Parameters for this Scenario

`MobileEngine.UM.SingleUserMode=true`

`MobileEngine.UM.SAPLogonTicketSupport=true`

`Mobile Engine.UI.CloseBrowserWindowSupport=true` (optional)

In this scenario users only need to enter a user ID and password when they log onto the system issuing the ticket. The logon data is verified in SAP MI using the SAP logon ticket. Password handling settings are consequently ignored in SAP MI and the user has no access to password management.

In the initial logon, which must be performed online, the user data from the logon ticket is used to create a user in the client.

Multiple Users

This scenario only applies to JSP clients.

The device is used by multiple users. The user starts SAP MI on their mobile devices as a service running in the background without a user interface. There must be an empty file named `startasservice.txt` in the same directory as the file `mobileengine.exe`.

To work with SAP MI, the user opens the SAP MI user interface from a link (for example, in SAP Enterprise Portal).

If the ticket does not exist, the user can start the SAP MI from the browser under the configured address, usually **`http://localhost:4444/index.htm`**, and log on with user ID and password. The system uses settings already in SAP MI for handling passwords and the user can use password management in SAP MI.

Before a user can use a SAP logon ticket, a user ID and password must be created for this user in the client.

Parameters for this Scenario

`MobileEngine.UM.SingleUserMode=false`

`MobileEngine.UM.SAPLogonTicketSupport=true`

Prerequisites

- The server (SAP Web AS) is configured to support SAP logon tickets, see Authentication and Single Sign-On [External].
- The client (SAP MI Client Component) is installed on the mobile device.

- If you want to use the multiple user mode, you have to have the JSP version of the client.
- A Win32 operating system or PDA operating system is installed on the mobile device. The latter is only supported by AWT clients.

Procedure

For information about the procedure for configuring the client with parameters in file `MobileEngine.config`, see Predefining and Setting Parameters for All Users [Page 36]

You must add or adjust the parameters described below if you want to support single sign-on. The parameters and values that are relevant for your scenario are described in the above scenario description.

Checking and Adjusting Existing Parameters

For JSP clients only:

1. In the system directory, check the entry under `\drivers\etc\hosts` and adjust it if necessary.
2. In the file `MobileEngine.config` enter the local host specified under `\drivers\etc\hosts` for the parameter `MobileEngine.Runtime.Host`.

   The system issuing the ticket is server *dnt123.abc.def.corp*.

   Windows 2000 is installed on the mobile device.

   Make sure that the following is specified under `<Drive>\WINNT\system32\\drivers\etc\hosts`:

   ```
   <IP address> localhost localhost.abc.def.corp
   ```

   Enter the following for parameter `MobileEngine.Runtime.Host`:

   `MobileEngine.Runtime.Host= localhost.abc.def.corp`

JSP and AWT clients:

- In the file `MobileEngine.config` check the relevant parameters `MobileEngine.Sync.Gateway, MobileEngine.Sync.Client, MobileEngine.Sync.Language` for the connection and adjust them if necessary.

Adjusting SSO-Specific Parameters

Define the parameters listed in the following table to activate and configure single sign-on.

Relevant Parameters for Single Sign-On

Parameters	Description
MobileEngine.UM.SingleUserMode	Activate use by a single user only. Possible values: **`true`**, **`false`** Default: **`false`** The setting for this parameter can be changed at runtime. If you change the value from **`false`** to **`true`**, the client checks if more than one user is registered on the device. If multiple users are registered, the client automatically resets the value to **`false`**.

Parameters	Description
MobileEngine.UM.SAPLogonTicketSupport	Activate single sign-on. Possible values: **`true`**, **`false`** Default: **`false`** The client can be used when Single Sign-On is activated in single user mode only when an online connection exists.
MobileEngine.UM.SAPLogonTicketBackgroundRequest	Activate background request (JSP clients only). This parameter defines if the logon ticket is requested without user interaction. With **`false`** user interaction is supported. Possible values: **`true`**, **`false`** Default: **`false`**
MobileEngine.UM.SAPLogonTicketRequestURL	Address from which the logon ticket was requested. Possible values: URL Default: None
MobileEngine.UM.SAPLogonTicketWaitingRefresh	Time in seconds after which the system again checks if the logon ticket was received (JSP clients only). Possible values: Integer values greater than 0 Default: **`3`**
MobileEngine.UM.SAPLogonTicketRequestTimeout	Time in seconds between requesting the logon ticket and cancellation. Possible values: Integer values greater than 9 Default: **`90`**
MobileEngine.UM.SAPLogonTicketLogoffURL	Address for the logoff from the system issuing the ticket (optional). Possible values: URL Default: None
Mobile Engine.UI.CloseBrowserWindowSupport	The option *End* is displayed on the homepage of the SAP MI (JSP clients only). Possible values: **`true`**, **`false`** Default: **`false`**
MobileEngine.UM.ExternalAuthUserParameter	Name of parameter used to transfer the user ID to and from the ticket-issuing system (AWT clients only). Possible values: any character string Default: None

Parameters	Description
MobileEngine.UM.ExternalAuthPasswordParameter	Name of parameter used to transfer the password to and from the ticket-issuing system (AWT clients only). Possible values: any character string Default: None
MobileEngine.UM.ExternalAuthAdditionalParameters	Parameters used to transfer additional information to the ticket-issuing system (AWT clients only). Possible values: any character string Default: None

Configuration of Framework Deployment

Use

The following sections contain information about configuring the framework deployment.

Activating BSP Application ME_FW_INSTALL

Use

With this procedure you can check if the *ME_FW_INSTALL* Business Server Page (BSP) application already exists and whether it has been activated. If this is not the case, you can create and activate it so that you can use it when installing the SAP MI Client Component.

Prerequisites

You activated all the required Basis services. For more information, see SAP Note 517484.

Procedure

1. Start transaction *SICF*.
2. Expand *default_host* → *sap* → *bc* → *bsp* → *sap*.
3. Check if the element already exists. If yes, click with the secondary mouse button on the element, check if the service was already activated, and choose *Activate Service*, if appropriate.
4. If the element does not yet exist, click with the secondary mouse button on the last expanded *sap* element and choose *New Sub-Element*.

   The *SAP namespace* dialog box appears.
5. Choose ✔.

   The *Create a Service Element* dialog box appears.
6. In the field *Name of service element to be created,* enter **`me_fw_install`** and select *Independent Service*.

7. Choose ✔.
8. In the field *Description,* enter a description of the service, for example, **`SAP Mobile Infrastructure Framework Installation`**.
9. Choose 💾.

   Service *me_fw_install* appears in the tree structure under *default_host* → *sap* → *bc* → *bsp* → *sap* and is not yet active.
10. To activate the service, click with the alternate mouse button on *me_fw_install* and choose *Activate Service*.

    The service is active (evident from the black font color).

Mobile Component Descriptor

Use

The Mobile Component Descriptor (MCD) describes the properties of an application, framework, or add-on that is to be used as a mobile component in the *SAP MI*. With an MCD you can also define the order in which the mobile components should be installed on the mobile device. See Defining the Installation Sequence [Page 96].

The developer of the framework, application, or add-on usually creates the corresponding MCD. It is shipped together with the installation file. If there is no corresponding MCD, you as a customer or consultant can also create it.

Existing MCDs are displayed in the SAP MI Web Console with *Upload Application*.

Features

The MCD contains the following required information:

Required Fields

Field	Meaning
Mobile Component	Technical name of the mobile component
Version	Version of the mobile component. The version has 1 -6 positions, is of type Character and can be defined as you like.
Version for Role-Based Assignment	Do not make this setting here. Instead, make it when uploading the component in the SAP MI Web Console. If there are different versions of the applications, the administrator can mark one version. The system then automatically uses this version when processing roles (see Assigning Mobile Components to a Role [Page 97]). There is no input field for the version when editing roles.
Description	Description of the mobile component.
Component Type	• **Application**: Mobile business application • **Framework**: Runtime environment for mobile applications • **Add-On**: Additional generic functions, for example, a DLL • **JVM**: Java Virtual Machine for PDAs • **SSL**: File for using the Secure Sockets Layer (see SAP

Field	Meaning
	Note **580497**) • **INSTALL_SEQUENCE**: Installation sequence defining the order for installation.
Runtime Environment	• **JSP**: If you want to upload Web-based applications. • **AWT**: If you want to upload Abstract Window Toolkit (AWT) based applications. • **MicroITS**: If you want to upload MicroITS-based applications.
Link to SAP MI Homepage	Choose **Yes**, if an application should appear as a link on the personal start page in the *SAP MI*. For all other component types choose **No**.

If you are an application developer, you can also define other settings if required.

Optional Settings

Tab Page	Description
SyncBO	SyncBOs that are needed by the mobile component and their properties. Here you can define whether application data should be visible for all users of a device and how to proceed if an error or conflict occurs.
Deployment	Path from which the files are downloaded at deployment. Do not make any entries here. The administrator must define the path for the download in the SAP MI Web Console.
Link	Other mobile components that are needed by this mobile component and that might need to be installed. When you define an installation sequence, you define the order in which the components should be installed on this tab.
Languages	Languages in which the mobile component is available.
Framework	Framework that must be installed in order for the mobile component to be installed. You can also define an area here.
Environ.	Conditions that must be satisfied for the mobile component to be installed.
Authorizations	Authorization objects needed by the user in order that the mobile components be installed.

Activities

With Transaction *MI_MCD* you can create new MCDs and search for MCDs by defining areas and by using * (asterisk) in order to edit these MCDs.

Once you have created a new MCD, you must upload the corresponding installation file from the SAP MI Web Console and make entries for certain fields. You can select one of the created MCDs there; see Uploading Framework Files [Page 67], Uploading Mobile Applications [Page 87] und Uploading Add-On Files [Page 90].

Starting the SAP MI Web Console

Prerequisites

If you want to launch the SAP MI Web Console with **`https`**, you configured the *SAP J2EE Engine* so that it can support SSL. You can find more information on the SAP Service Marketplace at **`service.sap.com/securityguide`**.

Administrators of the SAP MI Web Console need a user in the system with authorization for starting Transaction SMOMO.

If no data is displayed in the SAP MI Web Console, check that you have the required authorization; see Role Processing for Mobile Applications [Page 16].

Procedure

Launch the SAP MI Web Console with the following link:

http(s)://<Server>:<Port>/me/WebConsole/login

- Server: *SAP J2EE Engine* name
- Port: *SAP J2EE Engine* standard port For more information about the port definition, see SAP Note **590956**.

We recommend that you call the SAP MI Web Console with **`https`** to ensure that your data is transmitted securely.

Uploading Framework Files

Use

This procedure enables you to upload the SAP MI framework files to the SAP MI Web Console. The framework files contain the runtime environment for mobile applications.

Prerequisites

- You have saved the framework files you want in a folder of your choice.
- The Mobile Component Descriptor (MCD) is available in the system. If the MCD is not yet available and you have a suitable transport file, you can transport it to your system using transaction STMS. You can find information about transports in the documentation for the Transport Management System.

  For more information about MCDs see Mobile Component Descriptor [Page 84].
- You started the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose for the framework file you want to upload.

   Upload the required files:

   - MOBILEENGINE_AWT (only if you are using applications with runtime environment *AWT*), application type **`Framework`**

- MOBILEENGINE_JSP (only if you are using applications with runtime environment *JSP*), application type **`Framework`**
- JAVAVM (only if you are using a PDA), application type **`JVM`**

  The files for the various processors adhere to the following naming convention:

  `CrEme<Version>_<Processor>_<Version of operating system>_<Name of operating system>_minimal.CAB`

  - Version of CrEme: For example, **`324`** for Version 3.24 or **`325b13`** for Version 3.25b13
  - Processor: Prior to Version 3.25: **`XSC`** for XScale, PXA and others or **`ARM`** for Strong ARM; as of Version 3.25 **`AX`** for all processors
  - Version of operating system**`: CE 30`** for Pocket PC 2002 and Windows CE 3.x or **`CE4x`** for Windows Mobile 2003 or Windows CE.Net 4.x
  - Name of operating system: **`PPC`** for Pocket PC or **`HPC`** for Windows CE

  When using JVM files, note the comments under Prerequisites [Page 12].
- **SSL** (only if you are using the *SAP MI* security components). Application type **`SSL`**

  If you upload additional framework files for different platforms and languages, they are also called either MOBILEENGINE_AWT or MOBILEENGINE_JSP.

2. Enter the following data only:

Field	Purpose
Version for Role-Based Assignment	If there are different versions of an application, you can select one here. The system then automatically uses this version when processing roles (see Assigning Mobile Components to a Role [Page 97]). There is no input field for the version in the role editor.
Build Number	You can specify the build number of the application to establish at a later date which build of a mobile application you uploaded.
Use IP Instead of Host Name	Access to the host using the IP address. If the mobile devices cannot access the host on which the installation files are stored using the host name, set his indicator. In this case the IP address is used for the download link instead of the host name.
Domain in URL	If the mobile devices can only reach the host on which the installation files are stored using their fully qualified name in the network, enter the network domain in this field.
Protocol	Enter the protocol to be used when the files are downloaded to the mobile device, for example, **`http`** or **`https`**.

Field	Purpose
Application File	Choose the framework file in accordance with the language required, your operating system, and the selected runtime environment.  If the SAP MI Web Console is run in a J2EE cluster, the mobile components are stored on only a single node of the cluster. In this case you must manually copy the mobile components to all the other nodes of the cluster (see Copying Mobile Components to All Nodes of the J2EE Cluster [Page 94]).

For more information about the other fields, see Mobile Component Descriptor [Page 84]. You should only change it using the procedure described there.

You can add multiple installation files (such as Setup.exe for Win32, CAB files for WinCE) to the entry created here. The installation routine for the client (SAP MI Client Component) installation selects the installation file that is appropriate for the operating system.

Choose  in the line with the created entry. Specify the corresponding runtime environment, operating system and language again and select the file. Choose *Change*.

3. Choose *Change*.

Result

The framework files have been uploaded. Once the client has been installed on the mobile device, the framework files are visible in the SAP MI Web Console under *Administration* → *Administer Mobile Devices*.

 Configuration of Mobile Applications

Use

The following sections contain basic information about configuring mobile applications. For more detailed information about application-specific configuration, see the relevant application guides.

You have to perform the following configuration steps for each mobile application:

- Create the RFC destination in the backend (see Creating RFC Destinations Pointing to the Backend [Page 70]).
- Set up application in backend (see application guides)
- Make the Settings for Smart Synchronization [Page 74] or the Settings for Generic Synchronization [Page 72]
- Upload the application into the SAP MI Web Console (see Configuration of Deployment of Mobile Components [Page 84])

Creating an RFC Destination Pointing to the Backend

You create an RFC destination to the backend in the system in which the *SAP MI ABAP Server Component* is running. You need the RFC destination to provide the *SAP MI* methods, function modules, and synchronizer created in the backend in the *SAP MI ABAP Server Component*.

Create the RFC destination with the following naming convention: **<Backend-System-ID>CLNT<Client>**.

The RFC destination should have destination type 3 (connection to R/3 System).

The user specified under *Logon* must have authorization for all *SAP MI-specific* function groups contained in table *BWAFMAPP*.

For more information see Displaying, Maintaining and Testing Destinations [Page 70] and Entering Destination Parameters [Page 71].

Displaying, Maintaining, and Testing Destinations

To display, create, or change destinations, choose the following from the SAP menu: *Tools* → *Administration* → *Administration* → *Network* → *RFC Destinations* (transaction SM59).

Remote destinations are stored in the table RFCDES. The table RFCDES describes logical destinations for remote function calls (RFCs). You cannot maintain the table RFCDES directly.

Displaying Destinations

The initial screen shows a tree that allows you to select different connection types (such as partner systems or partner programs). For more information, see Destination Types [External].

To find a destination, choose *Find* and enter your selection criteria. A list of suitable entries is displayed. You can display all the available information for each entry.

Creating Destinations

On the initial screen for destinations, the connection types and all existing destinations are displayed in a tree structure.

For an explanation of the available connection types, see Destination Types [External].

To create a new RFC destination, choose *Create*. A new screen is displayed with empty fields in which you make entries.

If you are creating a remote destination, you can enter one particular application server, or a group of servers to distribute system load.

For more detailed information on destination parameters, see Entering Destination Parameters [Page 71].

Changing Existing Destinations

On the initial screen for destinations (transaction SM59), the connection types and all existing destinations are displayed in a tree structure.

To change an existing RFC destination, choose *Change*.

For more detailed information on destination parameters, see Entering Destination Parameters [Page 71].

Testing Destinations

To test a destination, choose the required function from the *Test* menu.

- *Connection* (also a pushbutton)
- *Authorization* (checks logon data)
- *Network of Application Servers* (returns a list of application servers)

Entering Destination Parameters

In addition to the *RFC destination*, you must enter the following information:

Technical settings

- *Connection type*

  Enter an existing connection type.
  For an explanation of the available connection types, see Destination Types [External].

- *Trace*

  Mark the *Trace* option to have the RFC communication logged and stored in a file. You can then display the file, both in the calling and receiving system, using report RSRFCTRC.

- *Load balance*

  If you choose load balancing, you must specify the following information:

  - *Target system* (For a list of available servers, log on to the target system and choose *Tools* → *Administration, Monitor* → *System Monitor* → *Servers*.)
  - *Message server* (Log on to the target system and choose *Control* → *Control Panel* from the CCMS main menu. It is the server that offers the service M.)
  - *Group* (of servers) (see SAP Logon Group of Servers)

  Otherwise, you must specify the following information:

  - *Target host*

    The name of a server host of the target system that you want to use as a port to the system.

  - *System number*

    Communications service used with the target system. To obtain it, choose *Tools* → *Administration* → *Monitor* → *System Monitor* → *Servers*.

Security Options

The following options are available only with some connection types:

- *Trusted system* (for type 3 only) If the target system is a trusted system, choose *Yes*. For detailed information about trusted systems, see Trusted/Trusting Relationships Between SAP Systems [External].

- *SNC* (Secure Network Communications, available for types 3 and T only) If you have an active SNC-supported security system, you can activate additional security options which you must set via *Destinations* → *SNC options*.

Description

Text description of the entry.

Logon

- *Language*

  System language to be used
- *Client*

  Client code
- *User*

  User name to be used for remote logon, if different from current user name
- *Password*

  User password
- *Current user*

  The current user name is to be used for remote logon.

The *Attributes* section contains creation and change information.

Destination Types [External]

Configuration of the Backend

Use

The configuration of the backend is application-specific. For more information, see the relevant application guides.

Settings for Generic Synchronization

Use

You can find information about the settings for generic synchronization below.

Changing Synchronization Container Processing

Use

You want certain synchronization containers to be processed directly in one of the back-end systems by RFC and not in the *SAP MI ABAP Server Component*. To do this, you must create RFC destinations on the level of method names in table *MEMAPPDEST*.

Prerequisites

You are using Generic Synchronization.

Procedure

Perform the following steps in transaction *SE16*.

1. List all the method names in the *BWAFMAPP* table in the back-end system.
2. List all the method names in the *BWAFMAPP* table from the *SAP MI ABAP Server Component*.
3. Table *BWAFMAPP* contains information about which function module belongs to a specific synchronization method (a method name). You can find some method names and their function modules in the *SAP MI Server Component*. Several method names, however, only exist in the back-end system.
4. **Only** those entries that exist in the back-end system and **not** in the *SAP MI Server Component* need be copied from table *BWAFMAPP* in the RFC back-end system to the table *BWAFMAPP* in the *SAP MI Server Component*.
5. You must create a corresponding entry in table *MEMAPPDEST* in the *SAP MI Server Component* for each copied method name. This ensures that the function modules that belong to the method names can be called in the back-end system with an RFC.

   You may **not** enter the following method names in table *MEMAPPDEST* since the corresponding function modules must be executed directly in the *SAP MI ABAP Server Component*:

   - WAF_REGISTRY
   - WAF_INSTALLATION_LOG
   - MEREP_REGISTRATION

   Also read the corresponding remarks in the installation guides for the mobile applications.

Meaning of the Fields in Table *MEMAPPDEST*:

Field Name	Purpose	Example
METHOD	Method from table *BWAFMAPP* in the back-end system.	CRMHH_SYNC_SELECT
RFCDEST	RFC destination that points to the back-end system where the BWAFMAPP entries can be found (see Creating an RFC Destination Pointing to the Backend [Page 70]).	CRMCLNT000
SYSRELEASE	Basis release of the back-end system.	610

Settings for Smart Synchronization

Use

The following sections contain information about the settings for Smart Synchronization.

Uploading a SyncBO

Use

With this procedure you can upload your SyncBO definitions from your local PC to the SAP MI ABAP Server Component.

Prerequisites

Your SyncBO definitions were downloaded to a file on your local PC.

Procedure

1. Start Transaction *merep_mig.*
2. Select *Upload from File* or *Import* for the transport action.

   Select transport action *Upload from File* for the migration of objects to a text file.
   Select transport action *Import* for the migration of objects as a transport file.
3. If you have choosen *Import* enter the transport request.
4. Choose *Simulate* for a test run. Choose *Execute* to start the upload.

   Choose *Simulate* to display the files to be uploaded.
5. Enter the name of an upload or transport file.
6. Choose *Open* to perform the import.
7. Confirm or change the RFC destinations.

   The RFC destinations should be defined in Transaction *SM59*.

The SyncBO definitions are downloaded.

See also:

See also Migrating SyncBOs Between Systems [External]

Defining RFC Destinations for SyncBOs

Use

By defining an RFC destination for a SyncBO, you can define the target backend system for synchronization with the SyncBO.

You can define either an RFC destination for each SyncBO or a default RFC destination to be used by all SyncBOs that do not define a specific RFC destination.

Prerequisites

An RFC destination was created for the target backend system and tested. The target backend system contains the application for the SyncBOs used by the mobile applications.

The RFC destinations should be defined using Transaction *SM59*, see Creating an RFC Destination Pointing to the Backend [Page 70].

During mobile data synchronization the RFC setting is used. The RFC user must have appropriate authorizations to execute the function modules in the backend system unless synchronization is performed using the Current User or in a Trusted System landscape.

Procedure

Defining a Specific RFC Destination for a SyncBO via the SyncBO Builder

1. Start Transaction *merep_sbuilder.*
2. Enter the SyncBO in field *SyncBO ID*.
3. Choose *Edit* → *RFC Destination*.
4. Choose *Yes* to confirm the message in the dialog box.
5. Select *SyncBO Specific* and select the RFC destination for the target system.
6. Confirm your entry by choosing *Continue*.

Defining a Specific RFC Destination for a SyncBO via the profile dialog

1. Start Transaction *merep_pd*.
2. Choose the *Synchronizer* tab.
3. Choose the *SyncBO* tab if not selected already.
4. Choose *Display <-> Change.*
5. Change the RFC destination of a synchronization control record of the corresponding SyncBO.
6. Choose *Save*.

Defining a Default RFC Destination

1. Start Transaction *merep_pd.*
2. Choose the *Synchronizer* tab.
3. Choose the *Destination* tab.
4. Choose *Display <-> Change* and enter the target RFC destination.

5. Choose *Save*.

Result

The SyncBOs with the specified RFC destination are synchronized with the given backend system. The SyncBOs without the specified RFC destination are synchronized with the default backend system.

Generating All Imported SyncBOs

Use

After uploading the SyncBO definitions, you need to generate the uploaded SyncBOs to enable them for actual use.

You need to perform this step every time you have imported new SyncBOs.

Generating the SyncBOs triggers the generation of ABAP function modules for each SyncBO. The generated ABAP code is executed for the data replication and synchronization.

This procedure enables you to generate multiple SyncBOs simultaneously.

Procedure

1. Start Transaction *merep_sbuilder*.
2. Choose *SyncBO* → *Generate Multiple SyncBOs…*.
3. Select your imported SyncBOs from the hit list and choose .

   The system confirms the successful generation.

Client Data Distributor

Use

The Client Data Distributor (program *MEREP_DISTRIBUTOR*) determines delta data independent of a download request from the client. The system creates and processes the worklists and stores relevant delta data in the outbox(es) for the defined devices. So the program triggers the same procedure as a client download request. This is independent of the SyncBO type.

The administrator can create variants for the program and define for which mobile components, mobile groups, users, devices, mobile IDs, or SyncBOs delta data should be determined and processed. The administrator can execute the variants manually or create a background job to execute them regularly.

If the applications use SyncBOs of type *Timed 2-Way* the execution of the program *MEREP_DISTRIBUTOR* should be scheduled after the expected end of the update of the replication database in order to achieve the highest available accuracy of data.

The feature can be used most efficiently if the following settings are defined:

- The client is configured to synchronize asynchronously.
- The client is configured not to send download requests in every synchronization trigger, especially not in the synchronization directly after the execution of the job.

Example

On the mobile devices there are mobile applications which use a combination of SyncBOs of type *Timed 2-Way* and other types. The background job for *Timed 2-Way* is scheduled to run every three hours between 5 am and 8 pm.

Most mobile devices are synchronized between 7 am and 9 am before sales people leave for there customers. This leads to an overload of the SAP MI Server component, as the system needs to determine delta data for each of the devices at this time.

To reduce the overload of the server, the administrator can schedule a background job to execute *MEREP_DISTRIBUTOR* at 6 pm. At this time the outboxes for the devices will be already filled when the devices start to synchronize at 7 pm.

Activities

To use this feature the following steps are necessary

- Creating variants for the program, see Creating Variants for the Client Data Distributor [Page 77]
- Define and configure a background job that executes the program, see Scheduling Background Jobs for the Client Data Distributor [Page 78]

Creating Variants for the Client Data Distributor

Use

To schedule a background job for the Client Data Distributor, you must define variants for the program *MEREP_DISTRIBUTOR*. The variants define for which devices the system determines delta data and creates worklists.

Procedure

1. Start Transaction *SA38*.
2. Enter **`MEREP_DISTRIBUTOR`** as program and choose *Execute*.
3. Define the devices for which you want to create a variant and choose *Save*.
4. Enter the name and a description for the variant you want to create.

   Keep the naming conventions for variants in mind (see Creating Variants [External]).
5. Choose *Save*.
6. Repeat steps 2 to 5 for each variant you want to create.

Scheduling Background Jobs for the Client Data Distributor

Use

You use background jobs to define when the system executes the variants of the program *MEREP_DISTRIBUTOR*. At the scheduled time the program determines delta data and creates worklists for the relevant devices.

Prerequisites

You created variants with which background jobs can be scheduled, see Creating Variants for the Client Data Distributor [Page 77].

Procedure

1. Start Transaction *SM36*.
2. Enter a name for the job under *Job name*.
3. Select a *Job class*.
4. Specify when the job should begin by choosing *Start condition*.
5. Choose *Immediate*.
6. Select the *Periodic Job* option and choose *Periodic value* to define how often it is repeated.
7. Select the corresponding period value and choose *Save* to save the period and to return to the *Start Time* screen.
8. Choose *Save* in the lower part of the *Start Time* screen to return to the *Define Background Job* screen.
9. Choose *Step* and then *ABAP program*.
10. In the *ABAP Program* section, enter **MEREP_DISTRIBUTOR** in the name field.
11. Enter a suitable defined variant.
12. Choose *Save* in the lower part of the *Create Step 1* screen to return to the *Define Background Job* screen.
13. Choose *Save* to save the completely defined job and pass it to the background processing system.

    Release the job so that it can be executed. No job, not even one that is scheduled for immediate processing, can be executed if it was not released.

Replicating Data from the Backend

Use

Application data from the backend system needs to be replicated to the SAP MI ABAP Server Component in order for the mobile device to receive up-to-date application data from the backend system.

This is not the case for SyncBOs with synchronization types *Upload* (*U01)* and *2-Way (S01)*. For synchronization type *Backend-Driven (T51)*, replication is triggered from the backend system through the application program.

Executing the replicator via transaction

1. Start Transaction *merep_ex_replic*.
2. Enter the SyncBO you want to run the replicator for in *SyncBO ID*.
3. Enter the required value for the *Log Level*.
4. Choose *Execute*.
5. The replication protocol is displayed.
6. Repeat steps 2 and 3 for each of your SyncBOs.

In the productive scenario, replication is run via scheduled jobs. To schedule the jobs, create variants for each of the SyncBOs (see Creating Variants for Replication [Page 80] and Scheduling Background Jobs for Replication [Page 80]).

Executing the replicator via the profile dialog

1. Start Transaction *merep_pd*.
2. Choose the *Synchronizer* tab.
3. Select the synchronizer control record you want to run the replicator for.
4. Choose *Execute replicator* for the SyncBO to run the replicator.

In the productive scenario, replication is run via scheduled jobs. To schedule the jobs, create variants for each of the SyncBOs (see Creating Variants for Replication [Page 80] and Scheduling Background Jobs for Replication [Page 80]).

Result

SyncBO data is retrieved from the backend system and the replication database is updated with the data. For every replication but the first, the replicator identifies delta data and updates them in the replication database.

Creating Variants for Replication

Use

To schedule a background job for an ABAP report program, you must define a variant.

Procedure

1. Start Transaction *SA38*.
2. Enter program **MEREP_REPLICATOR_START** and choose *Execute*.
3. Enter the *SyncBO ID* for which you want to perform replication and choose *Save As Variant*.
4. Enter the name and a description for the variant you want to create.

   Keep the naming conventions for variants in mind (see Creating Variants [External]).
5. Choose *Save*.
6. Repeat steps 2 to 5 for each SyncBO.

Scheduling Background Jobs for Replication

Use

You have to replicate data from the backend system to the SAP MI ABAP Server Component at regular intervals in order to make sure that the data in the SAP MI ABAP Server Component is up to date at all times.

Prerequisites

You created variants with which background jobs can be scheduled. For more information see Creating Variants for Replication [Page 80].

Procedure

1. Start Transaction *SM36*.
2. Enter a name for the job in field *Job Name*.
3. Select **A** for the job class.
4. Specify when the job should begin by choosing *Start condition*.
5. Choose *Immediate*.
6. Select *Period values* to define how often it is repeated.
7. Select the corresponding period value and then choose *Save* on the *Period Values* screen to copy the period and to return to the *Start Time* screen.
8. Choose *Save* in the lower part of the *Start Time* screen to return to the *Define Background Job* screen.
9. Choose *Step* and then *ABAP program*.
10. In the *ABAP Program* section, enter **MEREP_REPLICATOR_START**.

11. Enter the variant you defined.
12. Choose *Save* in the lower part of the *Create Step 1* screen to return to the *Define Background Job* screen.
13. Choose *Save* to save the complete job and pass it to the background processing system.

    Release the job so that it can be executed. No job, not even one that is scheduled for immediate execution, can be executed if it was not released.

For more information see Scheduling Background Jobs [Page 14] and Releasing Jobs [External].

Configuring Backend-Driven SyncBOs

To replicate application data using a SyncBO of type *Backend-Driven* (T51), you must first configure the following settings in the Mobile Component Descriptor (MCD):

- Mapping between the SyncBO ID and the object ID used by the backend (mandatory)

  The backend system uses its own object IDs. When the backend triggers the replication process to the server, the server maps the object ID transferred by the backend to the actual SyncBO ID.

- Delay with which the job is to be scheduled (optional; default: 300 seconds)

  The replication job is scheduled when the backend triggers the replication process to the server, using the interval (in seconds) defined for the SyncBO.

  If the interval is defined as 600, the job is scheduled 10 minutes (600 seconds) after the server receives the triggering event.

  For more information, see note 711983.

- Package size (optional; default: 5,000 rows)

  You must configure the package size (instead of using the default value) if one header record holds large numbers of item lines.

Procedure

Configuring the Object ID and the Interval

1. Start the Transaction *mi_mcd*.
2. Enter the application name and choose *Display MCD*.
3. Choose *Display* <-> *Change*.
4. Choose the *SyncBO* tab.
5. Enter the following settings for each SyncBO of type *Backend-Driven*:
   - The object ID in the *SyncBO-ObjectID* column.
   - The delay with which job is to be scheduled in the *Interval* column.
6. Choose *Save*.

   For general information on editing the Mobile Component Descriptor, see Mobile Component Descriptor [Page 84] .

Configuring the Package Size

Create the following entries in the table MEMSD_DEP:

Package Size Configuration

Field Name	Field Value
NAMESPACE	(initial)
NAME	**T51SYNCBOCONFIG**
VERSION	(initial)
TYPE	**SYNCBO**
TYPE_INDEX	**00000000001**
DEPENDENCY_NAME	**NAME**
DEPENDENCY_VALUE	**<SyncBO ID>** (e.g. **SAP_EXM001**)
NAMESPACE	(initial)
NAME	**T51SYNCBOCONFIG**
VERSION	(initial)
TYPE	**SYNCBO**
TYPE_INDEX	**0000000001**
DEPENDENCY_NAME	**PACKAGE**
DEPENDENCY_VALUE	**<rows>**, (e.g. **1,000**)

Alternatively you can change the value of the `ldf_package` parameter by using the SyncBO exit of the generated replication function module.

Configuring Synchronizer Control Records

Use

To synchronize data using the SyncBO, you need to activate it first. After the SyncBO is generated, you must configure the synchronizer control record. Then you can use the control record to activate the SyncBO.

For SyncBOs with synchronization types *Timed 2-Way (T01)* and *Backend-Driven (T51)* you need to execute their replicators once before you can activate them in the synchronizer control record. If the replicator has not been executed, you will not be able to activate the synchronizer control record. In this case the traffic light sign will be set to yellow.

If you trigger synchronization from a client device, the corresponding message for a SyncBO that is not activated, will not be processed. In the process log, a corresponding log message is recorded and status of the inbound worklist item will continue to be *I-Waiting*.

Procedure

1. Start Transaction *merep_pd.*
2. Choose the *Synchronizer* tab.
3. Choose *Display <-> Change*.
4. Make the following settings for each SyncBO:
   a. Select *Enabled*.
   b. Deselect *Push-enabled*.
   c. If you select *Filtered by Referencing SyncBOs* (*Ref.Filter*) this SyncBO is only downloaded via cascade download during the synchronization process of the referencing SyncBO. It will not be downloaded independent of the referencing SyncBO.
   d. Select *Check Type* (*T*). *(*Deselect this option for production to improve performance.)
   e. Enter **A** for *Save Data* (*S*). (**E** is sufficient for production.)
   f. Select *Use Handler* (*Hdlr*). (If you deselect this option, you can set the required log level for each SyncBO.)
   g. Select *BAdI Active* if you want to implement BAdI logic. To avoid performance loss do not select this flag if no BAdI logic is implemented. For information on implementing BAdI logic see note 600817.
5. Choose *Save*.

Creating a Mobile Group

Use

You can create mobile groups and assign them to registered mobile devices. For information about assigning mobile groups, see Assigning a Mobile ID to a Mobile Group [Page 83]

Procedure

1. Start Transaction *merep_pd.*
2. Choose the *Mobile Group* tab.
3. Enter a unique new group name (character or number) for the mobile group.
4. Choose *Create*.
5. Enter a description for the mobile group in *Short descr.*.
6. Select *Enabled*.
7. Choose *Save*.

Assigning a Mobile ID to a Mobile Group

Use

A mobile device can only be synchronized with the backend system using the Smart Synchronization framework if it is registered in the SAP MI ABAP Server Component.

You can register mobile IDs either manually or via the automatic deployment of applications to mobile devices. In that case, the mobile ID is created in the background and registered with the SAP MI Server Component through the standard MI deployment process.

To group mobile IDs and set filters easily you can use mobile groups. To use mobile groups you assign them to registered mobile devices.

Prerequisites

- A user was created in the SAP MI ABAP Server Component.
- You created a mobile group (see Creating Mobile Groups [Page 83]).
- You defined your user settings [Page 126] and left field *Device ID* empty.
- You synchronized your mobile device (see Performing Synchronization [Page 127]).

The mobile ID was automatically generated in the SAP MI ABAP Server Component and returned to the mobile device. Field *Device ID* in the settings on the mobile device is filled in (see Editing User Settings [Page 126]).

Procedure

1. Start Transaction *merep_pd.*
2. Choose the *Mobile ID* tab.
3. Use the input help for field *Mobile ID* to select the required mobile ID.
4. Choose  *Display <-> Change* and enter the mobile group you want to assign to the mobile ID.
5. Choose *Save*.

 Configuration of Deployment of Mobile Components

Use

The following sections contain information about configuring the deployment of mobile components.

 Mobile Component Descriptor

Use

The Mobile Component Descriptor (MCD) describes the properties of an application, framework, or add-on that is to be used as a mobile component in the *SAP MI*. With an MCD you can also define the order in which the mobile components should be installed on the mobile device. See Defining the Installation Sequence [Page 96].

The developer of the framework, application, or add-on usually creates the corresponding MCD. It is shipped together with the installation file. If there is no corresponding MCD, you as a customer or consultant can also create it.

Existing MCDs are displayed in the SAP MI Web Console with *Upload Application*.

Features

The MCD contains the following required information:

Required Fields

Field	Meaning
Mobile Component	Technical name of the mobile component
Version	Version of the mobile component. The version has 1 -6 positions, is of type Character and can be defined as you like.
Version for Role-Based Assignment	Do not make this setting here. Instead, make it when uploading the component in the SAP MI Web Console. If there are different versions of the applications, the administrator can mark one version. The system then automatically uses this version when processing roles (see Assigning Mobile Components to a Role [Page 97]). There is no input field for the version when editing roles.
Description	Description of the mobile component.
Component Type	• **Application**: Mobile business application • **Framework**: Runtime environment for mobile applications • **Add-On**: Additional generic functions, for example, a DLL • **JVM**: Java Virtual Machine for PDAs • **SSL**: File for using the Secure Sockets Layer (see SAP Note **580497**) • **INSTALL_SEQUENCE**: Installation sequence defining the order for installation.
Runtime Environment	• **JSP**: If you want to upload Web-based applications. • **AWT**: If you want to upload Abstract Window Toolkit (AWT) based applications. • **MicroITS**: If you want to upload MicroITS-based applications.
Link to SAP MI Homepage	Choose **Yes**, if an application should appear as a link on the personal start page in the *SAP MI*. For all other component types choose **No**.

If you are an application developer, you can also define other settings if required.

Optional Settings

Tab Page	Description
SyncBO	SyncBOs that are needed by the mobile component and their properties. Here you can define whether application data should be visible for all users of a device and how to proceed if an error or conflict occurs.
Deployment	Path from which the files are downloaded at deployment. Do not make any entries here. The administrator must define the path for the download in the SAP MI Web Console.

Tab Page	Description
Link	Other mobile components that are needed by this mobile component and that might need to be installed. When you define an installation sequence, you define the order in which the components should be installed on this tab.
Languages	Languages in which the mobile component is available.
Framework	Framework that must be installed in order for the mobile component to be installed. You can also define an area here.
Environ.	Conditions that must be satisfied for the mobile component to be installed.
Authorizations	Authorization objects needed by the user in order that the mobile components be installed.

Activities

With Transaction *MI_MCD* you can create new MCDs and search for MCDs by defining areas and by using ***** (asterisk) in order to edit these MCDs.

Once you have created a new MCD, you must upload the corresponding installation file from the SAP MI Web Console and make entries for certain fields. You can select one of the created MCDs there; see Uploading Framework Files [Page 67], Uploading Mobile Applications [Page 87] und Uploading Add-On Files [Page 90].

Starting the SAP MI Web Console

Prerequisites

If you want to launch the SAP MI Web Console with **`https`**, you configured the *SAP J2EE Engine* so that it can support SSL. You can find more information on the SAP Service Marketplace at **`service.sap.com/securityguide`**.

Administrators of the SAP MI Web Console need a user in the system with authorization for starting Transaction SMOMO.

If no data is displayed in the SAP MI Web Console, check that you have the required authorization; see Role Processing for Mobile Applications [Page 16].

Procedure

Launch the SAP MI Web Console with the following link:

http(s)://<Server>:<Port>/me/WebConsole/login

- Server: *SAP J2EE Engine* name
- Port: *SAP J2EE Engine* standard port For more information about the port definition, see SAP Note **590956**.

We recommend that you call the SAP MI Web Console with **`https`** to ensure that your data is transmitted securely.

Uploading the Database

Use

In order for the mobile components to be able to store data on the mobile device, you must install a supported database such as IBM DB2 Everyplace there **before** you install any further mobile components.

Prerequisites

- You have read Note **732737**.
- The SAP MI Client Component is installed on the mobile device.
- No mobile component was yet installed on the mobile device. If mobile components were already installed, you must uninstall them before installing the database.
- No mobile components were assigned to the mobile device.
- If you want to install the database from an installation sequence, install the database as the first in the installation sequence; see Defining the Installation Sequence [Page 96].
- If the mobile device has *Pocket PC 2002* as its operating system and you want to use encryption, the *Microsoft Encryption Pack* must be installed on the mobile device. See Note **728746**.

Procedure

1. Upload the suitable file as an add-on, depending on the operating system or processor; see Uploading Add-On Files [Page 90]. You can find the files in the *PocketPC* and *Win32* directories.

Mobile Device	With Encryption	Without Encryption
PDA with ARM processor or Intel Xscale processor	DB2e_ARM_encrypt.zip	DB2e_ARM.zip
Mobile device with Windows XP	DB2e_WinXP_encrypt.zip	DB2e_WinXP.zip
Mobile device with Windows NT or Windows 2000	DB2e_Win2000_NT_encrypt.zip	DB2e_Win2000_NT.zip

2. Give the add-on a name of your choice, for example DB2E.
3. Assign the add-on to the users; see Assigning Mobile Components to Users [Page 99].

   The database is deployed to the mobile device with the next synchronization.

Uploading Mobile Applications

Use

This procedure enables you to upload the installation files for mobile applications that are not yet available in the SAP MI Web Console. The uploaded files can be edited there.

Prerequisites

- You have obtained the installation files required for the mobile application and saved these to a folder of your choice in Windows Explorer. For more information about the storage location of the installation files, see the respective mobile application documentation on the *SAP Service Marketplace* at **`service.sap.com/instguides`**.

- The Mobile Component Descriptor (MCD) is available in your system. If the MCD is not yet available and you have a suitable transport file, you can transport it to your system using transaction STMS. You can find information about transports in the documentation for the Transport Management System.

  If there is no transport file for the MCD, you might have to create an MCD yourself (see Mobile Component Descriptor [Page 84]).

- You started the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose *Change* for the mobile component you want to upload.

   If the list does not contain an entry for the mobile component, first create an MCD for the component (see Mobile Component Descriptor [Page 84]).

2. Enter the following data:

Field	Purpose
Version for Role-Based Assignment	If there are different versions of an application, you can select one here. The system then automatically uses this version when processing roles (see Assigning Mobile Components to a Role [Page 97]). There is no input field for the version in the role editor.
Build Number	You can specify the build number of the application to establish at a later date which build of a mobile application you uploaded.
Use IP Instead of Host Name	Access to the host using the IP address. If the mobile devices cannot access the host on which the installation files are stored using the host name, set his indicator. In this case the IP address is used for the download link instead of the host name.
Domain in URL	If the mobile devices can only reach the host on which the installation files are stored using their fully qualified name in the network, enter the network domain in this field.
Protocol	Enter the protocol to be used when the files are downloaded to the mobile device, for example, **`http`** or **`https`**.
Application File	Use the Explorer to navigate to the storage location of the required file and select it. The selected operating system and language are of no importance since there is a file for each operating system and each language. If the SAP MI Web Console is run in a J2EE cluster, the mobile components are stored on only a single node of the cluster. In this case you must manually copy the mobile components to all the other nodes of the cluster (see Copying Mobile Components to All Nodes of the J2EE Cluster [Page 94]).

   For more information about the other fields, see Mobile Component Descriptor [Page 84]. You should only change these fields using the procedure described.

3. Choose *Change*.

   If you want to upload a new version of the application, you do not need to create a new entry. You can simply change the existing entry.

Choose  in the line with the entry to be changed. Change the version, enter the corresponding runtime environment and select the installation file. Choose *Change*.

Result

The application files have been uploaded. You now need to deploy the uploaded files on the mobile device (see Assigning Mobile Components To Users [Page 95]).

Defining User-Specific Data Filtering

Use

For each mobile application developed based on the *SAP MI,* you as application developer can define what data of an application each user can access (read, write, change). This data is filtered and provided for the user in the SAP MI Client Component.

User-specific data filtering is defined according to the SAP authorization concept, permitting data filtering to be managed with the authorization concept tools. Note that the security guidelines for *SAP MI* are valid here. For more information see Users and Roles (BC-SEC-USR) [External] and SAP Authorization Concept [External].

Prerequisites

- You imported the SyncBO *MIAUTH*. For information about importing SyncBOs see Uploading SyncBOs [Page 74]. To check if the SyncBO *MIAUTH* exists, start Transaction *merep_sbuilder*, enter **`MIAUTH`** and choose *Display*.

  You can see SyncBOs such as SyncBO *MIAUTH*, that are made available with a transport, in Table *MEREP_810* (Transaction *SE16*). This table also contains the valid transport numbers for performing a migration.

- SyncBO *MIAUTH* is active. To check this, start Transaction *merep_sbuilder*, enter **`MIAUTH`** and choose *Environment* → *SyncBO Profile*.

Procedure

Make sure that the authorization objects are linked with the corresponding mobile application. For each authorization object that is used by the application (see installation guide for the application), a dependency must be maintained in the Mobile Component Descriptor (MCD). If this is not the case, you can create the dependencies in the editor of the Mobile Component Descriptor on tab *Environment*, see Mobile Component Descriptor [Page 84]. Use the following dependencies:

Dependency	**Value**
DEPENDENCY_TYPE	AUTHOBJECT
DEPENDENCY_NAME	NAME
DEPENDENCY_VALUE	<Name of the authorization object>

Uploading Add-On Files

Use

This procedure uploads the installation files of the add-ons that contain additional generic functionality. The uploaded files can be edited there.

Prerequisites

- You have obtained the installation files required for the add-ons and saved them in a folder of your choice in Windows Explorer.
- The Mobile Component Descriptor (MCD) is available in your system. If the MCD is not yet available and you have a suitable transport file, you can transport it to your system using transaction STMS. You can find information about transports in the documentation for the Transport Management System.

  If there is no transport file for the MCD, you might have to create an MCD yourself (see Mobile Component Descriptor [Page 84]).
- You started the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose *Change* for the mobile component you want to upload.

   If the list does not contain an entry for the mobile component, first create a Mobile Component Descriptor (MCD) for the component (see Mobile Component Descriptor [Page 84]).
2. Enter the following data:

Field	Purpose
Version for Role-Based Assignment	If there are different versions of a mobile component, you can select a version here. The system then automatically uses this version when processing roles (see Assigning Mobile Components to a Role [Page 97]). There is no input field for the version in the role editor.
Build Number	You can specify the build number of the mobile component to establish at a later date which build of a mobile component you uploaded.
Use IP Instead of Host Name	Access to the host using the IP address. If the mobile devices cannot access the host on which the installation files are stored using the host name, select this field. In this case the IP address is used for the download link instead of the host name.
Domain in URL	If the mobile devices can only reach the host on which the installation files are stored using their fully qualified name in the network, enter the network domain in this field.
Protocol	Enter the protocol to be used when the files are downloaded to the mobile device, for example, **`http`** or **`https`**.

Field	Purpose
Application File	Use the Explorer to navigate to the storage location of the required file and select it. The selected operating system and language are of no importance since there is a file for each operating system and each language.  If the SAP MI Web Console is run in a J2EE cluster, the mobile components are stored on only a single node of the cluster. In this case you must manually copy the mobile components to all the other nodes of the cluster (see Copying Mobile Components to All Nodes of the J2EE Cluster [Page 94]).

For more information about the other fields, see Mobile Component Descriptor [Page 84]

3. Choose *Change*.

If you want to upload a new version of the mobile component, you do not need to create a new entry. You can simply change the existing entry.

Choose  in the line with the entry to be changed. Change the version, enter the corresponding runtime environment and select the installation file. Choose *Change*.

Result

The add-on files are uploaded. You now need to deploy the uploaded files on the mobile device (see Assigning Mobile Components To Users [Page 95]).

Add-ons are assigned to users in the same way as mobile components are assigned to users.

 Installing a Driver Add-on

Purpose

This process is intended for system administrators deploying a mobile application with peripheral requirements. The system administrator deploys drivers and connector add-ons that meet the requirements of the mobile application.

Prerequisites

There is a connector add-on (it can be deployed when the driver is deployed).

Process Flow

1. A mobile application with peripheral requirements is uploaded to the SAP MI Web Console.
2. The system administrator uses the driver selection tool [Page 94] to find the driver add-on that matches the target platform.

3. The administrator checks if the connector add-on for the target platform is uploaded to the SAP MI Web Console. If the matching driver and/or connector are not loaded to the SAP MI Web Console, they can be obtained from the *SAP Service Marketplace*.
4. After the connector and driver add-ons have been uploaded to the SAP MI Web Console, the administrator deploys them.

   If the connector is not installed on the target device, both the connector and the driver add-ons can be deployed at the same time as the application. If there is a connector in the target mobile device, only the driver and application need to be deployed.

Result

The mobile application and the required driver add-on are deployed to the target mobile device.

Uploading Driver Files

Use

With this procedure you can upload driver files.

Prerequisites

- You have obtained the required driver files and saved them in a folder of your choice in Windows Explorer. Information on the available drivers can be found in Note **761833**.
- The connector is installed on the mobile device.

  If the connector is not installed on the device, you can deploy it together with the driver files. The order of the installation on the device is not relevant. The procedure for uploading a connector corresponds to the procedure described here. Enter *Driver Add-On* as type.
- You started the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Select the *Upload Mobile Component* tab.
2. Select *Upload Mobile Component*.
3. Enter *Driver Add-On* as type.
4. Under *Application File,* enter the driver file by navigating to the storage location of the required file with the Explorer and selecting the file.

   If the SAP MI Web Console is run in a J2EE cluster, the mobile components are stored on only a single node of the cluster. In this case you must manually copy the mobile components to all the other nodes of the cluster (see Copying Mobile Components to All Nodes of the J2EE Cluster [Page 94]).

   The file contains information about the name and version of the driver. You cannot make any entries in those fields. The name and version are displayed in the list of mobile components once they have been created.
5. Also enter the following data:

Field	Purpose
Description	Enter a description.
Version for Role-Based Assignment	If there are different versions of a mobile component, you can select a version here. The system then automatically uses this version when processing roles (see Assigning Mobile Components to a Role [Page 97]). There is no input field for the version in the role editor.
Build Number	You can specify the build number of the add-on to establish at a later date which build of a mobile component you uploaded.
Use IP Instead of Host Name	Access to the host using the IP address. If the mobile devices cannot access the host on which the installation files are stored using the host name, set his indicator. In this case the IP address is used for the download link instead of the host name.
Domain in URL	If the mobile devices can only reach the host on which the installation files are stored using their fully qualified name in the network, enter the network domain in this field.
Protocol	Enter the protocol to be used when the files are downloaded to the mobile device, e. g. **`http`** or **`https`**.

For more information about the other fields, see Mobile Component Descriptor [Page 84]

6. Choose *Create*.
7. If you want to define additional settings, such as dependencies, you can do so in the Mobile Component Descriptor [Page 84]. Note that type *ADDON* is displayed here for driver files.

If you want to upload a new version of the driver, you do not need to create a new entry. You can simply change the existing entry.

Choose  in the line with the entry to be changed. Change the version, enter the corresponding runtime environment and select the installation file. Choose *Change*.

You can also make this modification in the MCD Editor. Note that type *ADDON* is displayed here for driver files.

Result

The driver files have been uploaded. You now need to deploy the uploaded files on the mobile device (see Assigning Mobile Components To Users [Page 95]).

Driver files are assigned to users in the same way as mobile components are assigned to users.

The drivers are device-specific, that is you cannot deploy a driver to every device. If the deployment fails for this reason, you get a corresponding message in the SAP MI Web Console.

You can configure the drivers using device configuration (see Configuration of Mobile Devices using Device Configurations [Page 101]).

Driver Selection Tool

Purpose

The Driver Selection Tool (DST) enables the SAP MI Web Console administrator to select peripheral driver(s) that meet the mobile application peripheral requirements. This selection process also considers the mobile application target device operating system, processor, virtual machine, and available transports.

Integration

The DST is integrated into the SAP MI Web Console.

Features

- **Display Matched Drivers**

  Displays available drivers that match the target OS, processor, VM, transport and application requirements.

- **Display Non-Matched Drivers**

  Displays available drivers that do not match the target OS, processor, VM, transport and/or application requirements. It also displays the first selection criteria that is not met.

Constraints

- The DST only recommends driver add-ons registered in the DST driver catalog. The catalog is updated via a Service Pack installation.
- The DST only displays the first reason for a mismatch. It may display the operating system, processor, virtual machine, transports, or attributes in that order. On two instances more information is presented:
  - Transports - If the reason for a mismatch is the transports, the DST will display all the transports that did not match.
  - Attributes - If the reason for a mismatch is the attributes, the DST will show all attributes that were not matched. For each attribute, if more than one option does not match, the DST will display only the first option that did not match.

    Refer to Driver Requirements Document Editor [External] for an example of attributes and options.

Copying Mobile Components to All Nodes of the J2EE Cluster

Use

If the SAP MI Web Console is run in a J2EE cluster, the mobile components are only stored on a single node of the cluster when they are uploaded to the SAP MI Web Console. As a result, the files might not be found when they are downloaded. In this case you must therefore copy the files manually to all the other nodes of the cluster

Prerequisites

You uploaded the mobile components to the SAP MI Web Console.

Procedure

1. On the SAP Web AS, open folder *pub* in directory J2EE in the document folder of the Web service of the SAP MI Web Console.

   Directory of the J2EE:

   ```
   D:\usr\sap\<SID>\<Instance name>\j2ee
   ```

   Document folder of the SAP MI Web Console:

   ```
   cluster\server0\apps\sap.com\com.sap.ip.me.webconsole\servlet_jsp
   \me\root
   ```

   Example for the entire directory:

   ```
   D:\usr\sap\C11\JC00\j2ee\cluster\server0\apps\sap.com\com.sap.ip.
   me
   ```

2. Copy folder *pub* entirely to all the other nodes of the cluster.

Assignment of Mobile Components to Users

Use

When assigning mobile components to users you have the following options:

- If you need to assign a large number of mobile components on the mobile devices in a certain order, you can simplify the installation process by defining an installation sequence (see Defining the Installation Sequence [Page 96]).
- By default, assignment is by roles. This makes it easier for you to manage mobile components and users (see Assigning Mobile Components to the Users of Roles [Page 97]).

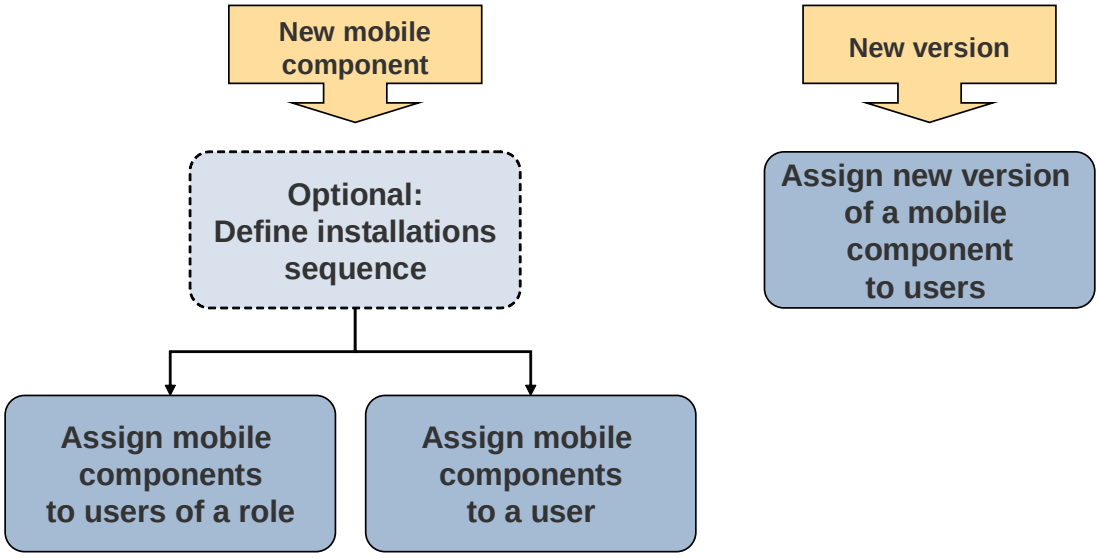
  We recommend that you use a role including all mobile users when you apply patches. In this way you can automatically upgrade all users to the latest patch level.

  You can define the version of the mobile component when you upload it (see Uploading Mobile Applications [Page 87]). You define that this version should be the default version for role-based assignment. This version is used for the users of all the roles to which the mobile component is assigned.

  You can also define the version if you assign mobile components for individual users, and not roles.

- You can also assign them manually if you would like to do so in certain cases (see Assigning Mobile Components to Users [Page 99]).
- If you already assigned your users previous versions of a mobile component, assign the users a new version of a mobile component (see Assigning Users a New Version of a Mobile Component [Page 100]).



Defining the Installation Sequence

Use

With an installation sequence you can define the order in which mobile components are installed on the mobile devices. The installation sequence is a separate application that controls the installation on the mobile device. The definition of an installation sequence is especially important for the first installation of a mobile device.

You define the installation sequence with the Mobile Component Descriptor Editor (MCD Editor). You can define the mobile components for which there is already a Mobile Component Descriptor (MCD). You cannot specify any other installation sequences.

Once you have defined the installation sequence, you assign it to a role or a user like a mobile component.

Note the following rules for defining and assigning the installation sequence:

- You can define one installation sequence only for each mobile device.
- The installation sequence can only contain one version of a mobile component.
- If one of the components referenced in the installation sequence is already installed on a mobile device, the installation sequence cannot be assigned to this device.

You cannot change installation sequences that are already being used. When you cancel the assignment of an installation sequence to a device, the applications installed with the installation sequence are not deleted from the mobile device.

Prerequisites

There must already be an MCD for each mobile component that you want to define in the installation sequence.

Procedure

1. Start transaction *MI_MCD*.
2. Choose *Create New Component*.
3. Enter a name and description for the installation sequence.

4. Choose *Create*.
5. Choose *INSTALL_SEQUENCE* as the component type.
6. On the *Link* tab page, add the mobile components you want to include in the installation sequence.
7. In the *Installation Sequence* column, enter the order in which the mobile components should be installed on the mobile device.
8. If required, enter the framework to be installed on the *Framework* tab page so that the installation sequence can be deployed.
9. Save your entries.

   The MCD for the installation sequence is now displayed in the SAP MI Web Console.

Assign the installation sequence to the users. You have various options here; (see Assigning Mobile Components to Users [Page 95]).

If you assign the installation sequence with a role, note the following:

- All the applications contained in the installation sequence must be defined as a default for the role in the SAP MI Web Console.
- The role must contain all components.

If you assign the installation sequence to a single user, the referenced components are also automatically assigned to the user.

Assigning Mobile Components to Users of a Role

With the following procedures you can assign mobile components to all the users of a role:

Assigning Mobile Components to Roles [Page 97]

Starting Role Synchronization [Page 107]

Assigning Components to Roles

Prerequisites

A MiniApp with **the same technical name** as the mobile component and **the type Offline Application** was created. The identical names link the mobile component with the MiniApp. The MiniApp is here a tool for assigning the mobile component to a role.

You can find more information about MiniApps in the Help Portal.

Procedure

You can manage your role either in the backend or on the *SAP Web AS*. If the majority of the functions of the mobile component that you want to assign to a role are running in the backend, we recommend that you manage the roles in the backend. If the majority of the functions are running on the mobile device, it makes sense to manage the roles on the *SAP Web AS*.

You can define the version of the mobile component when you upload it; see Uploading Mobile Components [Page 87]. You define that this version should be

the default version for role-based assignment. This version is used for the users of all the roles to which the mobile component is assigned.

You can also define the version if you assign mobile components for individual users, and not roles.

If you want to assign an installation sequence to a certain role, you must also assign all the components of the installation sequence to the role. For example, if your installation sequence contains three components, the role should also contain these three components and the installation sequence.

When Using a Backend with Releases 6.10 to 6.40 / Role Administration on the SAP Web AS:

1. Start transaction *PFCG*.
2. Enter the technical name of the role in field *Role* and choose  .
3. Choose the *MiniApps* tab and enter for *MiniApp* the technical name of the MiniApp that is identical with the technical name of the mobile component or installation sequence you want to assign to the role.
4. Choose  .

When Using a Backend with Release 4.6B or 4.6C:

1. Start transaction *PFCG*.
2. Enter the activity group (Release 4.6B) or role (Release 4.6C) you want to change.
3. Choose  .
4. Choose *Goto* → *MiniApps*.
5. Select *New Entries* and enter the following data:

Field	Description
Title	Any text
Height in Pixel	Any number
URL	Enter the URL **`http://localhost:4444/scripts/wgate/<Application>/!`**. <Application>: Technical name of the application or installation sequence

6. Choose  .
7. Proceed as described in SAP Note **521595**.

When Using a Backend with Release 4.5B:

In Release 4.5B you cannot assign a mobile component or installation sequence to a role because the concept of MiniApps does not exist in this release.

A workaround is to create a role on the *SAP Web AS* and assign mobile components or installation sequences to this role (see Chapter *When Using a Backend with Release 6.10 or 6.20 / Role Administration on the SAP Web AS*).

Starting Role Synchronization

Use

During role synchronization, the system transfers the assignment of the mobile components, installation sequences, and device configurations that you assigned to the users of a role to the SAP MI Web Console. Entries are added for newly assigned installation sequences, mobile components and device configurations. The entries for installation sequences, device configurations, and mobile components for which the assignment was canceled are deleted.

Prerequisites

You started the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose the tab page *Administration* → *Manage Mobile Devices*.
2. Choose *Start Role Synchronization*.

Field *Device ID* of the mobile components that are added contains the entry **on `all devices`** and is deployed on all the mobile devices of the user.

If you do not see the new entries, choose *Start* in the line containing the filtering information.

Alternatively, you can start the report *WAF_DEPLOYMENT_FROM_ROLES* (see Reports for Scheduling Background Jobs [Page 13]).

Result

The mobile component, installation sequence or device configuration has been saved and has the status *Install with Next Synchronization.* The next time the user synchronizes his mobile device, the mobile component, installation sequence or device configuration is automatically installed or executed on the device.

Assigning Mobile Components to Users

Use

This procedure enables you to assign mobile components to users who needs them for their daily work.

If you want to assign multiple mobile components at the same time and would like to define the order in which they should be installed, you can use an installation sequence. In this case you no longer need to assign the mobile components referenced in the installation sequence. For information about defining an installation sequence, see Defining an Installation Sequence [Page 96].

Prerequisites

You are in the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose the tab page *Administration* → *Manage Mobile Devices*.
2. Choose *Add Entries*.

   A new screen area appears.
3. Enter data as required.

   Leave the field *Device ID* **empty** and mark *On All Devices*. The specified mobile component or installation sequence is deployed to all the user's mobile devices.

   If you only want to deploy the given mobile component or installation sequence to one particular mobile device, enter only one device ID. This, however, should be the exception.
4. Choose *Add*.

   The new entry is listed in the overview. If you added an installation sequence, the installation sequence and the referenced mobile components are displayed in the overview.

Result

The mobile component or installation sequence is saved and has status *Deploy with the Next Synchronization*. The next time that the user synchronizes the mobile device, the mobile component or installation sequence is deployed on the mobile device automatically.

Assigning a New Version of a Mobile Component To Users

Use

This procedure enables you to assign a new version of a mobile component to a user.

Prerequisites

You are in the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose the tab page *Administration* → *Manage Mobile Devices*.
2. Choose to delete the entry with the old version of the mobile component.

   Make sure you remove the entry with device ID **`On All Devices`**.

   The mobile component is assigned status *Delete with Next Synchronization*.
3. Choose *Add Entries*.

   For more information, see Assigning Mobile Components To a User [Page 99].

Displaying the Status of Mobile Components

Use

In the SAP MI Web Console, you can track the status of the mobile components on a mobile device. Every time a mobile device is synchronized, it sends its current status (that is, its registry) to the SAP MI Web Console.

For example, if a new mobile component has been deployed on the mobile device, the status of the Deployment Console is one synchronization cycle behind the current status of the device. The success or failure of the deployment of the mobile component is only reported during the next synchronization of the device, when it then sends its current registry.

Prerequisites

You are in the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose the tab page *Administration* → *Manage Mobile Devices*.
2. Set a filter to display the entries you want to see.

   For more information about configuring the mobile device, select the device ID of the mobile device.

Result

The current status of a mobile component is displayed under *Status*.

Status Meaning:

Status	Meaning
Deploy with Next Synchronization	The mobile component is deployed on the mobile device during the next synchronization.
Delete with Next Synchronization	The mobile component is removed from the mobile device during the next synchronization.
Deployment Activated	The mobile component is currently deployed on the mobile device. The success message is expected during the next synchronization.
Deployment Successful	The mobile component has been successfully deployed on the mobile device.

Configuration of Mobile Devices using Device Configurations

Purpose

You can use this process to configure mobile devices. You define:

- The behavior of SAP MI Client Component on mobile devices
- The drivers installed on the mobile device

- The time and date settings on the mobile device

Process Flow

1. With transaction *DEVICE_CONFIG* you define monitoring configurations that contain the required parameters. For more information, see Defining Monitoring Configurations [Page 103].
    - For the configuration of the SAP MI Client Component, select function *MI_CONFIG* and enter the parameters and values that you want to set on the mobile device. For information about the parameters for the configuration of the SAP MI Client Component, see Predefining or Setting Parameters for All Users [Page 36]. (For parameters that are not available here you can use the process Configuration of Mobile Devices Using the Configuration File [External].)
    - Select *MI_PIOS* for the driver configuration. For information about configuring the drivers, see Driver Configuration [Page 103] and Note 761833.
    - Choose *TIME_AGENT* for the configuration of time and date settings (see Time and Date Settings [Page 105]).
2. Define device configurations for the various mobile devices and users with transaction *DEVICE_CONFIG*. On the *Assignments* tab page, define which configurations are assigned to which roles (see Defining Device Configurations [Page 106]).
3. After the assignment on the basis of roles has been completed, and if there are changes to the device configuration, start the role synchronization (see Starting Role Synchronization [Page 107]).

   If you want to assign the device templates to individual devices or users instead, you can do so in the SAP MI Web Console. In this case, proceed as for assigning applications (see Assigning Mobile Components to a User [Page 99]).

Result

The devices of the affected users receive the new device configuration with the next synchronization. The configuration is performed on the mobile device automatically. The user does not have to carry out any other additional steps.

Example

You want to define a given role that uses time-controlled synchronization and that is synchronized at intervals of one minute (60,000 milliseconds).

To do so, define a new monitoring configuration with transaction *DEVICE_CONFIG* , for example named *TimedSync*. Define a row for each parameter to be configured. For each parameter, select *MI_CONFIG* as monitoring function.

- Select *TIMEDSYNC* once as parameter ID and enter **`true`** as the value in order to activate timed synchronization.
- Select *TIMEINTERVAL* as the parameter ID for the other row and enter **`60000`** as the value in order to set the interval to 60,000 milliseconds.

Then create a new device template with transaction *DEVICE_CONFIG*. Enter the new monitoring configuration *TimedSync* on the tab page *Monitoring Functions*. You can also define additional monitoring configurations.

On the *Assignments* tab, also enter the role to which you want to assign the device configuration and the included monitoring functions.

Once you have performed the role synchronization (see Starting Role Synchronization [Page 107]) and synchronized the relevant users, timed synchronization with an interval of one minute is applied for all users of the role.

Defining Monitoring Configurations

Use

With monitoring configurations you can define various parameters for the SAP MI Client Component and for drivers installed on the mobile device. You can combine multiple parameters in one monitoring configuration.

Prerequisites

1. Start transaction *DEVICE_CONFIG*.
2. Define a meaningful name for the new monitoring configuration and choose *Create*.
3. Create a row for each parameter to be configured and define a value for this parameter.
4. Save your entries.

Result

You created a monitoring configuration that you can deploy to the mobile device using device configurations.

See also:

For more information on the process, see: Configuration of Mobile Devices using Device Configurations [Page 101]

For more information on alternative configuration options, see: Configuration of Mobile Devices [External]

For more information about the possible parameters for the configuration of the SAP MI Client Component, see Predefining or Setting Parameters for All Users [Page 36].

For more information on settings, see:

Driver Configuration [Page 103]

Time and Date Settings [Page 105]

Defining Data Carrier Synchronization [Page 46]

Configuring Repetitive Synchronization

Configuring Data Packaging [Page 49]

Configuring Data Compression [Page 50]

Configuring the Reset Option for the Logon Password [Page 51]

Configuring the Bypass Option for the Logon Password [Page 51]

Driver Configuration

Purpose

Peripheral input output services (PIOS) drivers can be configured by modifying several parameters. Parameters are defined for different configurations, and configurations are divided into configuration types. The system administrator can modify configuration parameters with the Mobile Infrastructure configuration system.

Implementation Considerations

Parameters of driver add-ons are formed with four tokens. The first token is the *driver name*. The second token is the *configuration type*. The third and fourth tokens are *configuration*

name and *parameter* respectively. Modification of driver parameters should follow the naming convention presented below:

<driver name>.<configuration type>.<configuration name>.<parameter>=value

Tokens and values are case-sensitive. The correct name must be entered in the MI configuration system to change a parameter value.

Integration

- Driver configurations are handled by the MI configuration system.
- Drivers and driver configurations are assigned using the SAP MI Web Console.

Features

Drivers have several parameters that can be used to change options for a driver. These parameters are defined for configurations that are separated into configuration types. PIOS drivers use the driver configuration type (cfg) to store parameters that modify how drivers connect to peripherals. This configuration type is also used to store parameters that are specific from driver to driver. A different configuration type is used to store the font configuration parameters (fntcfg) for peripheral type “Printer”. This configuration type (fntcfg) is used to configure fonts supported by the driver.

System administrators can modify driver configuration and font configuration parameters with the MI configuration system.

Constraints

- Adding a font with the font configuration parameters does not install the font on the physical printer. Printer fonts must be installed manually on the printer and should match the configured parameters.
- Configuration values are applied to drivers without validation. Unexpected behavior may be detected if a driver is not configured properly.

Example

Examples for parameter configuration and font configuration parameters are given below:

- In this example, a driver configuration parameter for the piprsymmf4t (Symbol microFlash 4t) printer driver is configured. This line sets the serial port baud rate to 9600 bits per second:

  `piprsymmf4t.cfg.Serial.BaudRate=9600`

- This example modifies a font configuration parameter for the piprmswin32 (Microsoft Windows 32-bit) printer driver. This line sets the “bitmapped bold italic” font options to bold and italic:

  `piprmswin32.fntcfg.BitmappedBoldItalic.Options=bold,italic`

See Also

You can find a list of all available parameters in a SAP note that is created for each driver. For a list of available drivers see the collective SAP Note **761833**.

Time and Date Settings

Use

You use this function to monitor and change the time and date settings on the mobile device. The server calculates the valid time for the mobile device based on the configured time zone and UTC time.

Prerequisites

- You have assigned the mobile device to a particular time zone.

Features

You have the following options:

Parameters for TIME_AGENT

Parameters	Description	Possible values
AGENT_MODE	Activate or deactivate monitoring	**ON, OFF**
AGENT_HISTORY	Activate or deactivate logging of battery capacity	**TRUE, FALSE**
AGENT_HISTORY_INTERVAL	Define intervals for logging (in minutes)	**<minutes>**
AGENT_REPORT_ON_FIRST_CYCLE or AGENT_REPORT_ON_SECOND_CYCLE	Define synchronization cycle for sending data if the device is used by multiple users. Set at least one of the parameters.	**TRUE, FALSE**
AGENT_TRIGGER_ON_SYNC_COUNT Or	Define the number of synchronization cycles run before data is sent to server.	Any integer **0** - Never **1** – With every synchronization **2** – With every second synchronization
AGENT_TRIGGER_ON_SYNC_INTERVAL and AGENT_TRIGGER_ON_SYNC_INTERVALTYPE	Determine the intervals (type and interval) with which the data is sent to server.	Interval type: **M** - Minutes **H** - Hours **D** - Days Interval: Any integer **0** - Never

Parameters	Description	Possible values
		1 – In intervals of one minute, one hour, one day.

Activities

You define the values for the parameters in a monitoring configuration (see Defining Monitoring Configurations [Page 103]). Use the TIME_AGENT function to monitor the time and date settings.

Defining Device Configurations

Use

With device configurations you can configure mobile devices without being able to access them directly. Device configurations can be distributed to the mobile devices with the usual deployment mechanism.

Prerequisites

You defined at least one monitoring configuration (see Defining Monitoring Configurations [Page 103]).

Procedure

1. Start transaction *DEVICE_CONFIG*.
2. Enter a name for the new device configuration and choose *Create*.
3. On the *Monitoring Functions* tab, enter the required monitoring configurations.
4. On the *Assignments* tab, enter the role you want to assign to the device configuration and thus to the monitoring functions.

   If you want to assign the device templates to individual devices or users instead, you can do so in the SAP MI Web Console. In this case, proceed as for assigning applications (see Assigning Mobile Components to a User [Page 99]).

Result

Once you have synchronized the roles in the SAP MI Web Console (see Starting Role Synchronization [Page 107]) the specified parameters are changed for all users of the given role with the next synchronization.

See also:

For more information on the process, see: Configuration of Mobile Devices using Device Configurations [Page 101]

For more information on alternative options to configure mobile devices, see: Configuration of Mobile Devices [External]

Starting Role Synchronization

Use

During role synchronization, the system transfers the assignment of the mobile components, installation sequences, and device configurations that you assigned to the users of a role to the SAP MI Web Console. Entries are added for newly assigned installation sequences, mobile components and device configurations. The entries for installation sequences, device configurations, and mobile components for which the assignment was canceled are deleted.

Prerequisites

You started the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Procedure

1. Choose the tab page *Administration* → *Manage Mobile Devices*.
2. Choose *Start Role Synchronization*.

Field *Device ID* of the mobile components that are added contains the entry **on `all devices`** and is deployed on all the mobile devices of the user.

If you do not see the new entries, choose *Start* in the line containing the filtering information.

Alternatively, you can start the report *WAF_DEPLOYMENT_FROM_ROLES* (see Reports for Scheduling Background Jobs [Page 13]).

Result

The mobile component, installation sequence or device configuration has been saved and has the status *Install with Next Synchronization.* The next time the user synchronizes his mobile device, the mobile component, installation sequence or device configuration is automatically installed or executed on the device.

Mobile Device Installation With an Installation Image (Optional)

Purpose

With the installation toolkit, the administrator can create an image containing an installation file. The mobile device user can then use this image or the installation file it contains to install the *SAP Mobile Infrastructure* on the mobile device. The end user can obtain the image from, for example, a CD or DVD, or from the network.

This procedure has the advantage that the user need not download all the mobile components and application data to the mobile device from the Internet. This process is, therefore, particularly useful in situations where the Internet connection is not stable or where the bandwidth is too low.

Two scenarios are possible when creating an installation image:

- Single users: The administrator creates an installation image for a known user, who for example wants to install the *SAP MI* on a new device because the old device is defective.
- Group users: The administrator creates an installation image that can be used by multiple users, and not only by a single user.

Prerequisites

- You set up a server for the installation toolkit, see Setting Up a Server for the Installation Toolkit [Page 109].
- The applications to be installed are synchronized with Smart Synchronization.
- The virtual users used to create the image have all necessary authorizations for synchronization.

Process Flow

Group User Scenario

The administrator creates the installation image for multiple unspecified users. To do this, the administrator creates a virtual user with which the administrator defines the data and components to be contained in the installation.

1. The administrator creates an installation image with the installation toolkit and assigns it to the virtual user.

   The virtual user must already exist in the R/3 System. Otherwise the administrator must create this user.
2. The administrator defines what should be installed for the virtual user in the SAP MI Web Console.
3. The administrator generates the installation image that contains an installation. If necessary, he or she adds batch files that perform actions before and after the installation.
4. The administrator burns the data on CD or DVD and gives it to the user.
5. The user installs SAP MI on the mobile device from CD or DVD. Alternatively, the administrator can install SAP MI, for example, if the user does not have the authorization to do this.
6. The user creates a user on the mobile device and logs on for the first time using this. The user must exist on the server (SAP MI Server Component).
7. The user starts the first synchronization.

Single User Scenario

The administrator creates an installation image for a specific user.

1. The administrator creates an installation image for a given user with the installation toolkit.
2. The administrator creates the installation image, burns it on CD or DVD, and passes it on to the user.
3. The user or administrator installs SAP MI on the mobile device using a CD or DVD.
4. The user creates a user and logs on for the first time with this. The user must exist on the server (SAP MI Server Component).

Setting Up Servers for the Installation Toolkit

Use

To create installation images you need to set up a separate server for the installation toolkit to run on. The installation toolkit consists of two parts: A Web application and a PC part. The Web application is already contained in the installation of the J2EE Engine. To set up the PC part of the installation toolkit you also need to run through the following steps:

- Providing additional files
- Creating the master instance

  The images created by the server are based on an empty installation of the SAP MI Client Component. This is used as a template for the images (the master instance).
- Providing DB software for the images

Prerequisites

Before you set up the server, decide which drive you want to use. All the files, the master instance, and the database have to be installed on the same drive. If you want to use a drive other than `C:`, you have to configure the installation toolkit accordingly (see Configuring the Installation Toolkit [Page 110]).

The following must be installed on the computer that you want to use as the server for the installation toolkit:

- Windows 2000, Windows XP or Windows 2003 Server
- J2EE Engine
- Java JDK 1.4.2
- Database software MaxDB Version 7.5.0.27 or higher 7.5-Version. For more information on installation and configuration, see Note 765433.

  If a MaxDB has already been installed, for example, for the J2EE Engine, you must install an additional MaxDB for the installation toolkit.

Providing additional files

1. The additional files for the installation toolkit can be found on the server in the archive ITK.zip under `…\j2ee\cluster\server0\apps\sap.com\com.sap.ip.me.insttool\META-INF`. Unpack this archive.
2. Create the directory `IMAGES` on the *C:*\ drive. Copy the contents of the directory `itool\IMAGES` into the newly created directory `IMAGES`.
3. Copy the contents from `itk\scripts` to `C:\sapdb\programs\bin`. or the directory specified during the installation of the MaxDB.
4. Copy the file `sapdbc.jar` from the directory `<installation directory for MaxDB>\programs\runtime\jar` into the directory `\IMAGE\itool\lib`.

Creating the master instance

To create the master instance, you can use either the standard version of the SAP MI Client Component or a version you modified (see Preconfiguring on Windows32 Platforms [Page 35]).

The Installation Toolkit only supports the JSP version of the SAP MI Client Component.

1. Start the `setup.exe` file on the host that you use as the server for the ITK for the installation of the SAP MI Client Component.
2. Copy all files and subdirectories from the installation directory of the SAP MI to `C:\IMAGES\MASTER\SAP\MobileInfrastructure`.
3. Open the file `startup.bat` in the directory `C:\IMAGES\MASTER\SAP\MobileInfrastructure\` and replace the first line (beginning with `ME_HOME=`) with **`if not defined ME_HOME set ME_HOME=`**.
4. If a patch exists for this version of the SAP MI Client Component, unpack the patch archive and replace the file MEg.jar in the directory `C:\IMAGES\MASTER\SAP\MobileInfrastructure\lib` with the new file from the patch archive.
5. If required, add further JAR files.
6. If you added JAR files in step 3 or if the file structure you copied to directory `\lib\` already contains JAR files, adjust file `listOfJars.txt`:
   a. Open file `C:\IMAGES\MASTER\SAP\MobielInfrastructure\listOfJars.txt` with `Notepad.exe` or another text editor.
   b. Replace all paths of form `C:\Program Files\SAP\MobileInfrastructure\lib\` with **`%ME_HOME%\lib`**.

      All the JAR files used by the SAP MI must reside in this folder.
   c. Add an entry with the form **`%ME_HOME%\lib\<Name of JAR file>`** for each additional JAR file added in step 4.
7. Save your changes in file `listOfJars.txt`.

Providing DB software for the images

1. Copy all the installation files for the maxDB software (the contents of the folder containing file `sdbinst.exe`) to directory `C:\IMAGES\TEMPLATE\Client\SAPDB`.
2. If the file `IASDB.TGZ` exists in directory `C:\IMAGES\TEMPLATE\Client\SAPDB`, delete it.

Configuring the Installation Toolkit

Use

If you have set up the server for the Installation Toolkit on a drive other than `C:` you must configure the Web application for the Installation Toolkit in accordance with that. You can configure the Web application for the Installation Toolkit by adjusting the file `ejb-jar.xml`.

Procedure

1. Open the file `META-INF\ejb-jar.xml` from `insttool.jar` with a text editor. You can find `insttool.jar` under `..\j2ee\cluster\server0\apps\sap.com\com.sap.ip.me.insttool\EJBContainer\applicationjars\`
2. Enter the required value under `<env-entry-value>`.

<env-entry-name>	<env-entry-value>
`itool/SAPDB/DRIVE`	Drive, for example, **`/D`**
`itool/SAPDB/SAPDB_PROG_DIR`	Path to bin directory of database, for example, **`D:\sapdb\programs\bin`**
`itool/SAPDB/SAPDB_DATA_DIR`	Path to data directory of database, for example, **`D:\sapdb\data\wrk`**
`itool/SAPDB/SAPDB_DEPENDENT_PATH`	Path to software installation directory for second instance of database, for example, **`D:\sapdb\mi`**
`itool/SAPDB/SAPDB_INDEP_PROG_PATH`	Path to software installation of database, for example, **`D:\sapdb\programs`**

3. Save your entries.
4. Create a new archive for the Installation Toolkit and use the Software Deployment Manager to deploy the data for the changed application (for more information, see Software Deployment Manager [External]).

Starting the Installation Toolkit

Procedure

Start the installation toolkit using the following link:

http(s)://<Server>:<Port>/mi/WebConsole

- Server: Name of J2EE Engine
- Port: Standard port of J2EE Engine, standard setting is 50000. For more information on determining the port, see the SAP Note **590956**.

Creating an Image for the Installation

Use

With this procedure you can create images for the installation of SAP MI on mobile devices.

Prerequisites

- You set up a server for the installation toolkit (see Setting Up a Server for the Installation Toolkit [Page 109]).
- The applications to be installed are synchronized using Smart Synchronization.
- You have started the installation toolkit (see Starting the Installation Toolkit [Page 111]).

- You started the SAP MI Web Console (see Starting the SAP MI Web Console [Page 86]).

Creating an Installation Image

1. In the installation toolkit, create a new installation with *Add New Image*.
2. Enter the following data:

Adding a New Image

Field	Purpose
Image Name	Enter a unique name that was not yet used for the image. The maximum length of the name is eight characters.
Type	Select *Single User* as the type if you want to create an image for a given user. Select *Collective User* if you want to create an image in which the user is not specified.
Technical User	Name of the virtual user created for the image (type *Collective User*). If you create an image of type *single user*, enter the name of the user here.
Password	Password of the technical user.
Framework	Framework to be installed with the image.
Version	Version of the framework.
System	Name of the system with which you want to synchronize.
Host	Name of the host on which the synchronization servlet is running.
Client	Client in the system with which you want to synchronize.
Port	Port number under which the synchronization servlet is running.
Protocol	Protocol for communications; possible value: http
Trace	Trace level to be used to create the image.
Delete Database Automatically	Here you can define if the database instance that is used to create the image is to be deleted automatically after completion.

3. Choose *Add* to start creation of the image.

With *Cancel* you go back to the overview of the image.

Creation of the initial image is completed when the image has status *Initial* and a device ID.

Defining the User Profile

1. Go to the SAP MI Web Console.
2. Assign applications to the newly created user.

Creating the Installation Image

1. Choose *Synchronization* in the installation toolkit to add the data and applications assigned in the SAP MI Web Console to the installation image.
2. Enter the password of the assigned technical user again.

3. Repeat step 1 until all the applications and data are there.
4. In the installation toolkit, select *Generate* for the installation image you want to create.
5. Enter the password of the assigned technical user again.
6. Choose *Generate* to start image creation.

   With *Cancel* you go back to the overview of the image.

   The system creates a setup directory in the background. You can find the files in directory `C:\IMAGES\<Name of the Installation Image>\Client`.

   Creation is completed when the image has status *Finished*.
7. If required, configure steps to be performed before or after the installation:
   - If you want to perform actions **prior** to the installation, create a file named *PreInstallation.bat* in the directory containing file *setup.exe*. You can use this file, for example, to install the JAVA Runtime Environment (JRE).
   - If you want to perform actions **after** the installation, edit file `PostInstallation.bat` in directory *\ExecutionAfterInstallation* and copy the relevant files to this directory. These files are copied to directory `%TEMP%\ExecutionAfterInstallation` during installation on the mobile device and executed after the installation.
8. Burn directory `Client` or the subdirectories and files it contains on CD or DVD.

   You can install the image on the mobile device without any user interaction (Silent Mode, see Installing Images on Mobile Devices without User Interaction [Page 119]).

See also:

Mobile Device Installation from the Installation Image (Optional) [Page 107]

Configuring the Installation Toolkit [Page 110]

Installing Images on Mobile Devices without User Interaction

Use

You can install an installation image on the mobile device without having to make user entries.

Requirements

You created an installation image.

Procedure

Start the installation file created during image generation with the following command:

```
start /w <Installation file> /s /v"/qn /L <Log file with path>"
```

```
start /w setup.exe /s /v"/qn /L C:\setup_itk.log"
```

Parameters for the Image Installation

Parameters	Purpose
/w <Installation file>	Starts the given installation file. The system waits until the process has finished.
/s	Starts the process without user interaction (Silent Mode)
/v“<Parameter> <Parameter>“	Passes additional parameters: /qn (User interface is not displayed) /L <Log file> (Log of the installation)

See also:

Mobile Device Installation from the Installation Image (Optional) [Page 107]

Creating the Image for the Installation [Page 111]

Setting Up SAP MI on the Mobile Device

As an administrator, you can install SAP MI on a mobile device in several different ways:

- If the mobile device has an Internet or network connection, you can download the installation file using a URL and install it directly using an assistant. See Installing the SAP MI Client Component from the Internet [Page 117] for more information.

  This can also be done by the user.

  If you do not want to use the assistant for the installation, you can use the parameters provided to start the installation file. See Parameters for Installation on the Mobile Device [Page 115].

- If you want to install without an Internet or network connection, you can use an installation image. See Installing the SAP MI Client Component From CD or DVD [Page 120] for more information. The image already contains applications and data for the user of the mobile device.

  This type of installation can also be done by the user.

- If you want to install SAP MI on several Compact Flash Cards, you have a number of options for single- and mass installations. See Installing SAP MI Client Component on Compact Flash Card [Page 120] for more information.

After the installation has been completed, the user of the mobile device must start the SAP MI and log on. See Starting and Logging Onto the SAP MI [Page 124]. To make sure the device can synchronize the user must edit the settings (if the installation is not already configured accordingly). See Editing User Settings [Page 126] for more information.

The user must then synchronize once or twice (depending on the type of installation used) to load all the applications and data onto the mobile. See Performing Synchronization [Page 127].

Changing the Initial Password

Use

Before you can install the *SAP MI* on your mobile device, you must define your own password for security reasons.

Procedure

1. Log onto the *SAP Web AS* on which the *SAP MI Server Component* is installed using your user and password **`INIT`**.
2. Assign a new password.

Parameters for Installation on the Mobile Device

You can control the installation of the SAP MI Client Component on the mobile device using installation parameters.

Port 4444 is used as the standard port for the Tomcat server. You can change the port, for example to prevent attacks by a worm at this port (see Note 677312). To change the port number to 5555, for example, start the installation as follows:

```
setup.exe /z“-tomcatport<5555>“
```

Syntax

```
Setup.exe /<Parameter>
```

The further syntax depends on the parameters.

Parameters for setup.exe

Parameter	Meaning
/r	Record a silent installation and write the entered values in a response file. Example: **`setup.exe /r`**
/s	Perform a silent installation in which no dialog boxes are displayed for the end user. Example: **`setup.exe /s`**
/f1	Alternatives: Specify the name and path for the response file. You can use this parameter both for recording and performing a silent installation The default name is `SetupIss.htm` Example: **`setup.exe /r /f1“C:\\temp\setup.iss“`**
/z	Specify additional options; see the table below. Example: **`setup.exe /z“-<Option><<Wert>> -<Option>“`**
/L	Specify the language for the installation and program; see the table below. The default value is the language of the operating system and, if this language is not available, English. Example: **`setup.exe /L1031`**

Additional Options (Parameter /z)

Option	Meaning
-path	Installation directory for the SAP MI Client Component Example: **setup.exe /z"-path<C:\Programs\SAP MI Client Component>"**
-protocol	Protocol to be used by the SAP MI Client Component when communicating with the SAP MI Server Component. The value is entered in the configuration file *mobileengine.config* (parameter *MobileEngine.Sync.Gateway.Protocol).* Possible values: **http** or **https** Example: **setup.exe /z"-protocol<http>"**
-host	Host name of the SAP J2EE Engine. The value is entered in configuration file *mobileengine.config*. (Parameter *MobileEngine.Sync.Gateway.Host*).
-system	System name of the SAP Web AS The value is entered in configuration file *mobileengine.config*. (Parameter *MobileEngine.Sync.Gateway.System*). Example: **setup.exe /z"-system<ABA>"**
-port	Standard port of the SAP J2EE Engine The value is entered in configuration file *mobileengine.config*. (Parameter *MobileEngine.Sync.Gateway.Port*).
-tomcatport	Port of the Tomcat server The value is entered in configuration file *mobileengine.config* (Parameter *MobileEngine.Runtime.Port*). Default value: **4444** Example: **setup.exe /z"-tomcatport<5555>"**
-tomcathandlerport	Port where the Tomcat server is started and ended.
-sapdbname	Name for the SAP DB The value is entered in configuration file *mobileengine.config* (Parameter *MobileEngine.Persist.Jdbc.Drive, MobileEngine.Persist.Jdbc.DbUrl, MobileEngine.Persist.Jdbc.DbName*).
-client	Client in the *SAP Web AS.* The value is entered in configuration file *mobileengine.config*. (Parameter *MobileEngine.Sync.Client*)
-pold	The SAP Mobile Infrastructure is started with the old parser.
-cu	The name of the logged on user is read from file *C:\\saptemp\compuser.txt* and used to enter the user.

Possible Values for Parameter /L

Language	Value
Chinese (Simplified)	2052
Danish	1030
German	1031

Language	Value
English (USA)	1033
French (Standard)	1036
Italian (Standard)	1040
Japanese	1041
Korean	1042
Dutch (Standard)	1043
Polish	1045
Portuguese (Standard)	2070
Russian	1049
Swedish	1053
Spanish (Traditional)	1034
Czech	1029
Hungarian	1038

Installing the SAP MI Client Component from the Internet

Use

With this procedure, you install the client (*SAP MI Client Component*) on your mobile device. You can also install the client from a CD or DVD (see Installing the SAP MI Client Component From CD or DVD [Page 120]). To do this, the SAP MI administrator must create an installation CD or DVD.

You can find more information about installing **Linux** operating systems in SAP Note 754221.

Procedure

1. If you are using a proxy server, check your browser settings and configure them again if needed. The settings depend on the network infrastructure you are using.
   - To change the settings on Windows32 operating systems (Internet Explorer):

     Choose *Tools* → *Internet Options* → *Connections* → *LAN settings*.
   - To change the settings on Pocket PC 2002:

     Choose *Start* → *Settings* → *Connections* → *Connections* → *Work* → *Change* → *Proxy settings*.
   - To change the settings on Windows Mobile 2003:

     Choose *Start* → *Settings* → *Connections* → *Connections* → *Proxy settings*.
2. Enter the following URL in Internet Explorer on your mobile device: **`http://<Server>:<Port>/sap/bc/bsp/sap/me_fw_install/install.htm`**.

<Server>: SAP Web AS name

<Port>: SAP Web AS standard port

For more information about determining the server and port, see Determining the Server and Port [Page 119].

If an error message occurs, read SAP Note 434918.

The installation wizard home page appears.

3. If you want to use automatic device recognition, choose *Continue*. If your mobile device runs on another operating system, choose *Select Others*.

Depending on your operating system, you may need to specify the runtime environment.

4. Enter data as required on the installation wizard screens.

WinCE installation: You can select the *SSL* option irrespective of whether you want to install a Java Virtual Machine.

Win32 installation: See Configuration of Security [Page 55] and SAP Note **580497**.

5. On the *Download* screen, choose the links displayed one after another to download the installation files to your mobile device. Under certain circumstances, you only have to download one file.

   A *File Download* dialog box appears.

6. Choose *Save this program to disk* and then *OK*.
7. Choose any directory for the installation files and store the installation files there.
8. Run the installation files that you have just saved.

When Installing On PDAs:

The client is installed on your mobile device.

When Installing On Windows32 Operating Systems:

9. Another installation wizard opens.
10. Follow the instructions given on the screen.

    Once you have entered all the data in the installation wizard, the system installs the client on your mobile device.

Result

The installation program has created an entry for starting SAP MI on your mobile device.

If you are using a PDA, restart your mobile device. For all other mobile devices, restart SAP MI. For more information, see Starting and Logging Onto SAP MI [Page 124].

Determining the Server and Port

Procedure

1. Start Transaction SE80 in the system on which the *SAP Web AS* is running.
2. Enter the **`BSP application`** and **`ME_FW_INSTALL,`** and choose *Display*.
3. Expand *ME_FW_INSTALL* → *Pages with Flow Logic* and double-click on *Install.htm* to display the information for this object in the right window.
4. Choose tab *Properties*.
5. The server and port within the URL are displayed in field *URL*. You can use this URL.

Installing Images on Mobile Devices without User Interaction

Use

You can install an installation image on the mobile device without having to make user entries.

Requirements

You created an installation image.

Procedure

Start the installation file created during image generation with the following command:

```
start /w <Installation file> /s /v"/qn /L <Log file with path>"
```

```
start /w setup.exe /s /v"/qn /L C:\setup_itk.log"
```

Parameters for the Image Installation

Parameters	Purpose
/w <Installation file>	Starts the given installation file. The system waits until the process has finished.
/s	Starts the process without user interaction (Silent Mode)
/v"<Parameter> <Parameter>"	Passes additional parameters: /qn (User interface is not displayed) /L <Log file> (Log of the installation)

See also:

Mobile Device Installation from the Installation Image (Optional) [Page 107]

Creating the Image for the Installation [Page 111]

Installing the SAP MI Client Component From CD or DVD

Use

This procedure installs the *SAP MI Client Component* on your mobile device if your administrator gave you a CD or DVD with an installation image that was created with the Installation Toolkit.

If your *SAP MI* administrator did not give you an appropriate CD or DVD for the installation, you must perform the installation manually; see Installing the SAP MI Client Component from the Internet [Page 117].

Prerequisites

- Your administrator gave you a CD or DVD containing an installation image.
- You have administrator authorization for the mobile device.

Procedure

1. Place the CD or DVD in the corresponding drive.
2. Change to the CD or DVD in directory `\Client\setup` oder `\setup`.
3. Start file *setup.exe*.
4. For your installation directory, select a directory with as short a name as possible (for example C:\MI) as otherwise the installation might fail because the path name is too long.
5. The SAP MI Client Component and other applications and data are installed on the mobile device.
6. Restart the system.
7. Restart the mobile device, see Starting and Logging Onto the SAP MI [Page 124].

Result

The installation program has created an entry for starting SAP MI on your mobile device.

Installing SAP MI Client Component on Compact Flash Card

Use

You can install the SAP MI Client Component on the Compact Flash Card (CF card) of a PDA. If you want to perform the installation on multiple devices, you can first create a master installation. You can then either copy it manually to the mobile devices or create a new installation file with it.

Prerequisites

You determined the correct version of CrEme, see Uploading Framework Files [Page 67].

Procedure

Installing the SAP MI Client Component on the PDA

1. Deploy the framework files on the mobile device and install them there, see Installing the SAP MI Client Component from the Internet [Page 117].

   Do **not** perform a synchronization afterwards.

Installing a Patch on PDA

1. If there is a patch, copy the corresponding JAR file to directory *\MI\lib*.
2. Adjust file *\MI\listOfJars.txt*:
   a. Copy the files to a directory of your choice in which you can edit the file.
   b. Enter the path and name for the JAR file, for example

      `\MI\lib\MI25SP03patch02;<further JAR files>`.

      Separate the entries with a semicolon (`;`).
3. Copy the file back to the directory.
4. Adjust file *\MI\creme_listOfJars.txt* by specifying the path and name for `-classpath`, for example **`-classpath='MI\lib\MI25SP03patch02;<further JAR files>'`**.

Copying SAP MI to the PDA

Copy directory *\MI* to *\CF Card\MI*.

Installing Database Support

1. If you want to use applications that need database support on the mobile device, copy the relevant JAR file, for example *db2ejdbc.jar*, from the database add-on to directory *\CF Card\MI\lib*.
2. Adjust file *CF Card\MI\listOfJars.txt* as described above for patches.
3. Adjust file *\MI\creme_listOfJars.txt* as described above for patches.
4. Copy the relevant DLL files for the database to directory *CF CARD\creme\bin*:, for database DB2e for example:
   - *CryptoPlugin.dll*
   - *DB2e.dll*
   - *DB2eJDBC.dll*
5. Insert the following lines in file *CF CARD\MI\settings\MobileEngine.config*:
   - **`MobileEngine.Persist.Jdbc.PreparedStatementCacheSize=15`**
   - **`MobileEngine.Persist.Encryption=`**
   - **`MobileEngine.Persist.Jdbc.Driver=com.ibm.db2e.jdbc.DB2eDriver`**
   - **`MobileEngine.Persist.Jdbc.DbUrl=jdbc:db2e:%d`**
   - **`MobileEngine.Persist.Jdbc.DbName=db2e`**
   - **`MobileEngine.Persist.Jdbc.Db2.Dll=DB2eJDBC,CryptoPlugin, DB2e`**
   - **`MobileEngine.Persist.Implementation.TransactionManager = com.sap.ip.me.persist.jdbc.DBPersistenceManagerImpl`**
   - **`MobileEngine.Persist.Implementation.TransactionManager2 = com.sap.ip.me.persist.jdbc.DBPersistenceManagerImpl`**
   - **`MobileEngine.Persist.Implementation.DescriptorRuntime = com.sap.ip.me.persist.jdbc.DBDescriptorRuntimeImpl`**

- **MobileEngine.Persist.Implementation.PersistenceRuntime = com.sap.ip.me.persist.jdbc.DBPersistenceRuntimeImpl**
- **MobileEngine.Persist.Implementation.QueryRuntime = com.sap.ip.me.persist.core.QueryRuntimeImpl**

Modifying and Copying the Start Link (AWT Version)

1. For the AWT version, copy file *\windows\Start Menu\MobileEngine.lnk* to a directory of your choice in which you can edit the file.
2. Change:
   - `"\windows\CrEme\bin\CrEme.exe" to "\CF Card\CrEme\bin\CrEme.exe"`
   - `-cf 'MI\nlfont.txt' to -cf 'CF Card\nlfont.txt'`
   - `-cf 'MI\creme_listOfJars.txt to -cf ,CF Card\creme_listOfJars.txt'`
   - `'-home:/MI' to '-home:CF Card\MI'`

     Example for a modified start link; modified elements are in bold.

     ```
     255#"\CF Card\CrEme\bin\CrEme.exe" -Of -cf 'CF Card\MI\mlfont.txt' -cf 'Card\MI\creme_listOfJars-txt' com.sap.ip.me.core.Startup '-home:\CF Card\MI'
     ```
3. Copy the modified file back to directory *\windows\Start Menu*.
4. Copy file *MobileEngine.lnk* from directory *\windows\Start Menu* to directory *CF Card*.

Modifying and Copying the Start Link (JSP Version)

1. For the JSP version, copy file *\windows\startUp\MobileEngine.lnk* to a directory of your choice in which you can edit the file.
2. Change:
   - `"\windows\CrEme\bin\CrEme.exe" to "\CF Card\CrEme\bin\CrEme.exe"`
   - `-cf 'MI\creme_listOfJars.txt' to -cf 'CF Card\creme_listOfJars.txt'`
   - `'-home:/MI' to '-home:CF Card\MI'`

     Example for a modified start link; modified elements are in bold.

     ```
     255#"\CF Card\CrEme\bin\CrEme.exe" -Of -cf 'CF Card\MI\creme_listOfJars.txt' com.sap.ip.me.core.Startup '-home:\CF Card\MI'
     ```
3. Copy the modified file back to directory *\windows\Start Menu*.
4. Copy file *MobileEngine.lnk* from directory *\windows\startUp* to directory *CF Card*.

Adjusting Files on the CF Card

1. Adjust file *CF Card\MI\listOfJars.txt* as described above for patches. Change the paths so that they refer to the relevant directories on the CF card.
2. Adjust file *CF Card\MI\creme_listOfJars.txt* as described above for patches. Change the paths so that they refer to the relevant directories on the CF card.

Preparing for Mass Installation

1. On your computer, create a directory named *MobileOnCfCard*.

2. Copy all the files and subdirectories from directory *\CF Card* to this directory. The directory should have the following structure:

   CF Card

   CrEme

   MI

3. In directory *MobileOnCfCard* create a directory named *\windows\Start Menu* (AWT) or *\windows\startup* (JSP) and copy file *MobileEngine.lnk* to this directory.
4. Restart the device.

Manually Performing Mass Installation

1. Copy the subdirectories and files from directory *MobileOnCfCard* to the PDA.

   Subdirectories *CF Card* and *windows* must overwrite the corresponding directories on the device.

2. Restart the device.
3. Report these steps for PDAs with the same processor and operating system.

Performing Mass Installation Using the Installer

Alternatively to manual installation, you can create a new CAB file for the installation on the mobile device. To do this you need the Pocket PC Software Development Kit, see Modifying CAB Files for PDAs [Page 53].

1. Copy files *cabwiz.ddf*, *cabwiz.exe* and *Makecab.exe* to directory *MobileOnCfCard*.
2. Copy the files for the SAP Cab tool (see Note 746778) to directory *MobileOnCfCard*.
3. Open the input request and navigate to directory *MobileOnCfCard*.
4. Enter the following:

   ```
   java -jar cabTool.jar . out.inf "Mobile Infrastructure"
   ```

   The following message appears: `.inf File successfully created.`

5. Create the CAB file with the following entry.

   ```
   Cabwiz.exe out.inf
   ```

   File out.cab is created.

You can copy this file to all PDAs with the same processor and operating system and install it there simply by clicking. Alternatively, you can deploy the files to the mobile devices from the SAP MI Web Console.

Starting and Logging Onto the SAP MI

Starting the SAP MI

Starting SAP MI on a Windows32 Operating System

The SAP MI installer creates an entry in the Windows toolbar. By default, this is *Start* → *Programs* → *SAP Mobile Infrastructure*.

Select the *SAP Mobile Infrastructure* entry to start SAP MI.

The client (SAP MI Client Component) is started together with a browser that displays the SAP MI.

Starting SAP MI on PDAs

- **JSP Version**

  The installer creates a link in the Autostart folder of the operating system in order to start the SAP MI. The SAP MI therefore starts automatically when you restart your PDA.

  In the Windows start menu of the operating system, the installation program creates a link to the start of the Pocket Internet Explorer with a link to the SAP MI home page.

  If problems occur when you start up, keep the following in mind:

  On PDAs it is not possible to start the SAP MI Client Component from the address *<protocol>:///localhost:4444*. Instead, use the address *<protocol>://127.0.0.1:4444*.

- **AWT Version**

  The installer creates a link in the Windows starting menu of the operating system to start SAP MI.

  Start SAP MI by clicking on the link that was created.

Starting SAP MI on Linux Operating Systems

For more information, see SAP Note 754221.

Logging On

Enter the required logon data and choose *Log On*. If you do not yet have a user, create one by selecting *New User* and entering the required data.

If SAP MI is configured for single sign-on and you are using the device alone, you need not enter your user data, neither at the initial logon nor later on. Your user for SAP MI is created from the data in the SAP logon ticket.

If your device is configured for single sign-on and for use by multiple persons, you must first create a user before being able to use single sign-on. If there is no SAP logon ticket that can be used for the authentication, you must log on directly to the SAP MI with the user ID and password.

If your device is configured to bypass the logon password query (bypass function), you do not have to enter a password when logging on to SAP MI or

creating users. Your logon to the operating system is considered by SAP MI to be sufficient authentication in this case.

If the logon password query for your device is deactivated later, and you have already defined a logon password, when you make a change to the configuration you are asked to confirm the deactivation of your logon password. You do not have to enter a password to logon to SAP MI. You can also choose not to use the bypass option and to continue logging on using the password.

If you have forgotten your logon password, and you client has been configured appropriately, you can reset your password using the synchronization password. To do this, you require an online connection to the server.

If your logon password is identical to your synchronization password, and you cannot verify your ID on the server, the administrator must reset your synchronization password for you. Once this has been done, you logon to the server using the reset password and reset your logon password to this password. You can now logon to the client as normal and manage the passwords yourself.

Parameters for Starting the SAP MI

Use

With the start parameters you can control how you start the SAP MI Client Component on the mobile device and analyze any errors that occurred.

Syntax: **`mobileengine.exe -<Parameter>`**

Features

Parameters for mobileengine.exe

Parameter	Meaning
-a	Starts the AWT version of the SAP MI
-v	Activates verbose mode, which displays all the details about the current processes. This supports debugging.
-t	Activates trace mode to support debugging.
-f	Maximizes the window of the SAP MI after start-up
-s	Minimizes the window of the SAP MI after start-up
-?	Displays a help text
-h	Displays a help text
-help	Displays a help text

Example

```
mobileengine.exe -s
```

Editing User Settings

Use

The first time you start *SAP MI*, you must make some user settings. These settings are needed during synchronization.

Procedure

1. On the SAP MI start page, choose *Settings*.

   Note that the screen is different for the technology platforms AWT and JSP, for example some of the fields are not displayed on the screen for AWT.

2. Enter the following information:

Field	Entry/Meaning
User	User ID for your *SAP Web AS* system (is only displayed here and cannot be preconfigured, because in certain circumstances there could be more than one user working on one mobile device)
Client	Client for your *SAP Web AS* system
Language	Your preferred language
Country	Select the country in which or for which you work. The country has an effect on the currency, date, time, and number formats.
Time Zone	Define your current time zone
Daylight Saving Time	The system displays whether or not you have daylight saving time.
Gateway (is only displayed in the AWT version)	 You can optionally fill in the fields *Protocol*, *Host*, *Port* and *System* individually. <Protocol>://<Host>:<Port>/meSync/servlet/meSync?~sysid=<System>& • <Protocol>: Internet protocol (http or https) • <Host>: Host name of the SAP J2EE Engine • <Port>: SAP J2EE Engine standard port For more information about the port definition, see SAP Note **590956**. • <System>: SAP Web AS system name
Protocol	Select the required Internet protocol. The HTTPS protocol ensures secure data transmission.
Host	Host name or IP address of the J2EE Engine
Port	Standard port of the SAP J2EE Engine For more information about the port definition, see SAP Note **590956**.
System	SAP Web AS system name
Proxy	Set a flag if you want to make proxy settings

Field	Entry/Meaning
Host (proxy)	Your system administrator can tell you the values for the proxy settings.  You can find more information about the proxy settings for **Linux** operating systems in SAP Note 754221.  Proxy settings only come into effect after rebooting.
Port (proxy)	See *Host (proxy)*
Device ID	Device ID of the mobile device that was created by your system administrator. The *Device ID* field initially contains spaces and is automatically filled after the first synchronization.

3. Save your entries.

To activate and configure data packaging or to suppress the download of data to the mobile device, choose *Enhanced Settings*, see also Using Data Packaging [External] and Suppressing the Download [External].

Performing Synchronization

Use

You must synchronize the mobile device in order to make the mobile applications assigned to you available on your mobile device. In daily operations, you also synchronize to transfer new and changed data from the client (*SAP MI Client Component*) to the back-end system and vice versa.

If you cannot use online synchronization, you can use various data carriers such as memory sticks, diskettes and DVDs. In this case the system only synchronizes application data. The system can **not** install any new applications with data carrier synchronization.

Prerequisites

You have started the client on the mobile device and logged on.

If you want to synchronize using the data carrier, you must satisfy the following requirements:

- The administrator has activated data carrier synchronization (see Defining Data Carrier Synchronization [Page 46]).
- At least one mobile application is installed on the mobile device.

Synchronization with Online Connection

1. On the SAP MI start page, choose *Synchronize*.
2. If necessary, enter the synchronization password.

   The system synchronizes and displays a synchronization log.
3. Choose *Next*.

The system offers the mobile applications on your device and generates one link for each application.

4. Restart the device to activate the new mobile applications. You only have to restart the SAP MI on Windows32 operating systems.

   A link is now displayed on the start page of the SAP MI for each application. You can call the mobile application using this link.

Synchronization By Data Carrier

1. If you want to use data carrier synchronization, connect the data carrier with the mobile device or place it in the drive.
2. On the SAP MI start page, choose *Data Carrier Synchronization.*

Importing Data Provided by Administrator on Data Carrier

If you have not received the appropriate files from your administrator, you can skip the following two points.

3. Enter the synchronization files (file extension **.mis*) you want to read and choose *Continue.*
4. If you received the data on more than one data carrier, change the data carrier if necessary in order to read in further files.

Copying Synchronization Data for the Administrator on Data Carrier

5. Select a data carrier and start synchronization.

   The system creates one or more synchronization files with the file extension **.mis* in a suitable size for the data carrier.

6. If you have already received data from the administrator and imported it, and if you are using the same medium to transport the data, remove the synchronization files that were previously imported from the data carrier or change the data carrier.
7. Copy the created synchronization files to the data carrier. If you want to use a CD or DVD, copy the files to a temporary directory and then burn them on the appropriate data carrier.
8. Give the administrator the data carrier. The administrator can then synchronize it with the back end.

Additional Information

Uninstalling the SAP MI Client Component

Use

With this procedure, you can uninstall the client (*SAP MI Client Component*) on your mobile device.

Uninstalling on Windows 32 Platforms

1. End SAP MI on the mobile device by clicking the *SAP MI* icon in the Windows menu bar with the secondary mouse button and choosing *Exit*.
2. Choose *Windows Start* → *Programs* → *SAP Mobile Infrastructure* → *Uninstall*.

3. Follow the instructions given on the screen.

   The client is uninstalled.

   Under *Windows Task Manager* → *Processes*, monitor the process *java.exe*. If it takes too long until *java.exe* finishes or if the uninstallation program informs you that files to be updated are still being used, end *java.exe* in the Windows Task Manager.

Uninstalling on Pocket PC 2002 /Windows Mobile 2003

1. Delete the shortcut *MobileEngine.lnk/MobileInfrastructure.lnk* under *Start* → *Programs* → *File Explorer* → *My Device* → *Windows* → *AutoStart*.
2. Restart the mobile device.
3. Choose *Start* → *Settings* → *System* → *Remove Programs*.
4. Select and delete the following entries:
   a. *SAP AG ME/MI<Release>*
   b. *SAP AG, NSIcom CrEme...*
5. Manually delete files and folders that could not be deleted automatically in the *File Explorer*.
6. Restart the mobile device.

 Deleting Server Data for Mobile Devices and Users

Use

The SAP MI Server Component defines a device ID for each mobile device. The device ID and the corresponding data are also retained if the SAP MI Client Component is uninstalled or if the end device no longer exists. The data for users who no longer exist in the system is also retained in the SAP MI Server Component.

You can delete the data of mobile devices and users that no longer exist from the SAP MI Server Component.

After you deleted the data for a device, the device can no longer access the SAP MI Server Component. If you want the device to be able to synchronize again with the SAP MI Server Component, you or the end user must uninstall the SAP MI Client Component and then install it again; see Setting Up the SAP MI on the Mobile Device [Page 114]. A new device ID is created for the device at this time.

Prerequisites

The mobile device no longer exists or the SAP MI was uninstalled.

Activities

You can delete the data with function modules DELETE_ALL_DATA_OF_USER (data of a user) and DELETE_ALL_DATA_OF_DEVICE (data of a device). To do so, start Transaction *SE37* and execute the corresponding function modules.

Appendix A: Legal Statements of the Third Party Products

The Apache Software License, Version 1.1

Copyright (c) 1999-2001 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowlegement: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowlegement may appear in the software itself, if and wherever such third-party acknowlegements normally appear.
4. The names "The Jakarta Project", "Struts", and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their names without prior written permission of the Apache Group.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

SAP and/or its distributors may offer, and charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of the software. However, SAP and/or its distributors may do so only on its/their own behalf, and not on behalf of the Apache Software Foundation or its contributors.