

07.07.2005

JSR-179 Location API for J2ME™ - Change Log

#	Change Description	Reason	Spec version	Status
1	In the javax.microedition.location package description, the table listing the mapping between the permissions and API methods has been changed to include the method <code>LocationProvider.getLastKnownLocation</code> to the list of method protected by the <code>javax.microedition.location.Location</code> permission.	This method was missing from the table by error in the 1.0 release of the specification. The method description of the method described this already in 1.0, so this was an internal inconsistency in the spec and is now corrected.	1.0.1	PROPOSAL
2	<code>Coordinates.distance</code> method: the accuracy requirement has been relaxed from 0.35% to 0.36%.	The original intent was to allow simple spherical approximation to be used for calculations. It has turned out that some implementations of such algorithms may in some special cases provide results where the error is just slightly above 0.35%. To allow these approximations to be still used, the requirement has been relaxed by 0.01%.	1.0.1	PROPOSAL
3	Landmark class description, 2 nd paragraph has been updated to take into account that the objects are mutable and can be modified using the <code>set*</code> methods.	The 1.0 version erroneously claimed that the values could only be set by the constructor, which was inconsistent with the descriptions of the methods.	1.0.1	PROPOSAL
4	Landmark class description has been updated to clearly explain that the <code>QualifiedCoordinates</code> stored in the object is only a reference and not a copy of the originally passed in <code>QualifiedCoordinates</code> object. Semantics related to modifying these mutable objects have been clarified.	The 1.0 version was somewhat ambiguous on whether the <code>QualifiedCoordinates</code> is stored by reference or by copying. In order to have consistently behaving implementations, the behaviour is now clarified.	1.0.1	PROPOSAL
5	<code>LandmarkStore</code> class description has been updated to clarify the supported	Previously it was unclear which characters are	1.0.1	PROPOSAL

07.07.2005

	characters in landmark store names. It now requires that all other Unicode characters than the following list need to be supported: 0x0000...0x001F control characters 0x005C '\' 0x002F '/' 0x003A ':' 0x002A '*' 0x003F '?' 0x0022 '"' 0x003C '<' 0x003E '>' 0x007C ' ' 0x007F...0x009F control characters 0xFEFF Byte-order-mark 0xFFFF...0xFFFF	allowed. The list of excluded characters is compiled in order to allow direct implementations on top of common file systems.		
6	In LandmarkStore class description, a note has been added to further elaborate that the concept of a landmark “belonging to a LandmarkStore” is different than this landmark being currently stored in that LandmarkStore.	The specification text in 1.0 already specified this, but in a concise way that caused some misunderstandings initially. The further elaborated text makes it easier to understand the difference.	1.0.1	PROPOSAL
7	LandmarkStore.getInstance and deleteLandmarkStore methods clarified to explicitly state that the implementation is allowed to handle the store name parameter either as case-sensitive or as case-insensitive.	1.0 version was ambiguous and could have been implemented in either way. Because this depends on implementation strategy and possibly e.g. on the underlying file system, it was decided to explicitly allow both options.	1.0.1	PROPOSAL
8	LandmarkStore.createLandmarkStore method further clarified to say that it may in some cases modify the string passed in e.g. for differentiating between stores on different storage drives. A note has been added to strongly recommend the method to preserve character case of the store name.	1.0 specification already contained some statements specifying this behaviour, but it has been further elaborated to clarify the intended behaviour.	1.0.1	PROPOSAL
9	LandmarkStore.listLandmarkStores method: clarification that the strings returned by this method must be directly usable as input to getInstance or deleteLandmarkStore.	This was not explicitly stated in 1.0 version, although implicitly assumed. To avoid any misunderstanding, it's now	1.0.1	PROPOSAL

07.07.2005

	deleteLandmarkStore.	explicitly stated.		
10	LandmarkStore.updateLandmark exception conditions changed to allow throwing IllegalArgumentException if the store name parameter is longer than supported by the implementation.	The addLandmark method had this already in 1.0, but it was missing from updateLandmark.	1.0.1	PROPOSAL
11	LocationProvider.setLocationListener description has modified to explain that before the first valid location fix is obtained, the timeout parameter has no significance. After the first valid location fix is delivered to the listener, then the updates are resynced with the specified interval and the timeout parameter is used to allow updates to be delayed from the interval by the specified amount.	Previously the text about delivering invalid location updates before the first fix is obtained and the text about the timeout parameter could have been considered conflicting. Now this conflict is resolved.	1.0.1	PROPOSAL