

# Interoperability Testing

A SIPTECH CASE STUDY

Version: 1.03

Created Date: 21.Dec.2004

---

## Inside this document ...

01. Background
02. Problem / Challenges
03. Method / Intervention / Solution
04. Results Achieved
05. Lessons Learned / Conclusion



© 2005 All rights reserved. SIP Technologies and Exports Ltd. The information contained in this document is CONFIDENTIAL and PROPRIETARY in nature, and subject to the rights and ownership of SIPTECH. Any and all unauthorized copying or use of the contents hereof is prohibited.

## 1. Background

A leading mobile and wireless industry expert is developing a framework on the J2ME Connected Device Configuration, which provides a robust, flexible environment for applications running on mobile phones.

The platform is built using an incremental development strategy. Testing some of the components of this framework requires Interoperability Testing (IOT).

Interoperability Testing is complex and demanding, but indispensable. The high demands of IOT on overall system expertise and the need for effective integration of results and experiences into the quality assurance processes explain why few companies in the wireless market have the capabilities to benefit from IOT. A network manufacturer's or a mobile platform vendor's success in this discipline therefore represents a notable competitive advantage in maximizing the end-user's benefit and satisfaction.

## 2. Problem / Challenge

IOT testing is done, as part of the software's feature testing, to validate the implementation conformance with its specification for Instant Messaging and Presence Services and Data Synchronization which follows respective Open Mobile Alliance (OMA) specifications.

**Instant Messaging and Presence Services (IMPS)** - Used for exchanging messages and presence information between mobile devices, mobile services and Internet-based instant messaging services.

**Data Synchronization** - Used to back-up device's data such as Contacts, Calendar, and Email etc.  
The following needed to be achieved: -

- ❖ Testing and validating new features during development
- ❖ Regression tests in simulation environments
- ❖ Compliance testing with multiple servers
- ❖ Populating data at server side for data synchronization

### 3. Method / Intervention / Solution used

The product developer selected SIPTECH for doing IOT testing on a continuous basis, during and after the main phase of software development.

The strategy adopted by SIPTECH was:

- ❖ Develop IOT tests to validate functionality during the main phase of product development.
- ❖ Use the IOT tests to check the implementation's conformance with the specification.
- ❖ Perform End to End testing to completely validate the features of the sub-systems.
- ❖ Automate the IOT tests using a test tool for regression testing.

IOT testing for IMPS was done with multiple servers like FollowApp, Oz and Ericsson which are fully compliant with the Open Mobile Alliance (OMA) specifications for Instant Messaging and Presence Services. Multiple clients were created and messages were sent simultaneously to test the performance under conditions of stress.

IOT testing for Data Synchronization was done with multiple servers like FusionOne and NewMagically, which are fully compliant with the Open Mobile Alliance (OMA) specifications for Data Synchronization. Data synchronization was also done with multiple records to validate the component's performance under stress. Populating data at server side has been automated using the AutoMate5 tool from Unisys Software.

The focus is on specification conformance, correcting residual errors and reaching high robustness. SIPTECH has a well-defined coverage test methodology that is part of the overall test methodology applied successfully to the Software Testing Lifecycle.

### 4. Results Achieved

The following has been achieved:

- ❖ Tested the implementation interoperability against multiple servers.
- ❖ Tested the stability of the product under stressful dynamic test conditions.
- ❖ Tests developed were forward compatible. Tests developed for OMA specification v1.0 were also run against OMA specification v1.1.

## 5. Lessons Learned / Conclusion

- ❖ Test automation using AutoMate5.
- ❖ Measuring code coverage in frequent intervals helps to cover the uncovered areas in testing.
- ❖ Issues at the integration level can be minimized by testing the sub-systems individually.
- ❖ The interoperability verification works as a "compatibility test" in a real end-to-end environment for improving the maturity of the product before it is admitted to final evaluation by the customer.