



## CHAPTER 2

# Backward Compatibility

---

This release of JTAPI is backward compatible with applications written for release 5.1. Upgrading the CiscoJtapiClient is not mandatory. Applications must upgrade to 6.0 CiscoJTAPIClient only if they wish to use any of the new features introduced in release 6.0.

## Definition

Backward compatibility indicates that applications will continue to operate following a Cisco Unified Communications Manager software upgrade with the same level of functionality:

- Without any application changes
- Without making any Cisco Unified Communications Manager configuration changes

JTAPI clients (all APIs) are considered to be backward compatible if all the following conditions are satisfied:

- The existing feature behavior remains the same.
- The event set that is associated with the existing features remains the same.
- When a new feature impacts the behavior of an existing one, you can disable the new feature to maintain the behavior and event sets of existing features.

Changing the application configuration following an upgrade may impact backward compatibility.

After a Cisco Unified Communications Manager software upgrade, the JTAPI client may also require an upgrade.

To reduce the impact on existing interfaces when new features are introduced, applications should take advantage of the interface `CiscoCallEv.getCiscoFeatureReason()`, which returns the feature reason code. Applications should have a default implementation for unrecognized values for `getCiscoFeatureReason()`. As new features are introduced, applications should expect to see new reason codes.

JTAPI is backward compatible. This means that the current JTAPI client will work with the next release of Cisco Unified Communications Manager. However, the JTAPI client in the next release should not be expected to work with the current release. Applications must downgrade the JTAPI version whenever Cisco Unified Communications Manager is downgraded.

In some features, two or more calls get merged into a single call. For all JTAPI-supported features in which calls are merged, `CiscoCallChangedEv` gets delivered to the call observer. `CiscoCallChangedEv` also gets delivered for other features, such as transfer, conference, barge, cbarge, and unpark. Previous cause codes that are returned for `getCallControlCause()` have not changed.

# Guidelines

The following guidelines for applications will help achieve backward compatibility:

- Applications must gracefully ignore events that they do not handle. For example, applications can ignore CiscoCallChangedEv if the application chooses to use CiscoTransferStart and End events.
- Applications must have a default behavior for unrecognized reason codes.
- Applications must have a default behavior for undefined error codes.
- Applications should expect to see new values for getCiscoFeatureReason() when JTAPI adds new features in the future.
- Any behavior that is not defined in the JTAPI 1.2 specification and not described in the JTAPI Interface specification should not be considered a feature.

# Dedicated Features

The following features are designed to facilitate backward compatibility with future releases.

### getCiscoFeatureReason() on CiscoCallEv Interface

This interface provides CiscoFeatureReason in the JTAPI call event for current and new features. Older features, such as Transfer, will continue to receive the old CiscoCause that was provided by the previous interface, CiscoCallEv.getCiscoCause (), as usual. This interface provides REASON\_TRANSFER for transfer.

```
com.cisco.jtapi.extensions
Interface CiscoCallEv
```

```
int          getCiscoFeatureReason()
```

This interface returns CiscoFeatureReason.

### Public interface CiscoCallChangedEv

The public interface CiscoCallChangedEv extends CiscoCallEv. This event is delivered when two or more calls are merged into one. The transfer, conference, unpark, Barge, and CBarge features can cause this event to be delivered. In addition to the older interfaces on this event, getCiscoFeatureReason() returns the feature code that is defined in the CiscoFeatureReason interface.

### Public interface CiscoFeatureReason

JTAPI provides CiscoFeatureReason in call events caused by features. CiscoFeatureReason is provided for existing as well as new Cisco Unified Communications Manager features. For example, the TRANSFER and CONFERENCE feature reasons would be REASON\_TRANSFER and REASON\_CONFERENCE.



**Note** This interface will provide new reasons for new features introduced in the future, and will therefore not be backward compatible. Applications that use CiscoFeatureReason should expect to receive new reasons and must implement default behavior to maintain Application’s backward compatibility.

static int	<b>REASON_TRANSFER</b> Indicates events are due to the transfer feature.
static int	<b>REASON_FORWARDNOANSWER</b> Indicates events are due to the call forward no answer feature.
static int	<b>REASON_FORWARDBUSY</b> Indicates events are due to the call forward busy feature.
static int	<b>REASON_FORWARDALL</b> Indicates events are due to the call forward all feature.
static int	<b>REASON_REDIRECT</b> Indicates events are due to the call being redirected.
static int	<b>REASON_BLINDTRANSFER</b> Indicates events are due to the single-step transfer feature.
static int	<b>REASON_BARGE</b> Indicates the events are due to the call being barged.
static int	<b>REASON_CONFERENCE</b> Indicates events are due to the conference feature.
static int	<b>REASON_PARK</b> Indicates events are due to the call being parked.
static int	<b>REASON_CALLPICKUP</b> Indicates events are due to the call pickup feature.
static int	<b>REASON_NORMAL</b> Indicates the events are due to normal call activity.
static int	<b>REASON_PARKREMAINDER</b> Indicates events are due to the park remainder feature.
static int	<b>REASON_UNPARK</b> Indicates the events are due to the call being unparked.
static int	<b>REASON_IMMDIVERT</b> Indicates events are due to the idivert feature.
static int	<b>REASON_FAC_CMC</b> Indicates events are due to the forced access code and client matter code feature.
static int	<b>REASON_QSIG_PR</b> Indicates events are due to QSIG path replacement feature.

