

European Union bail-in rules: Implications for Canadian lenders and borrowers

MAR 28, 2017 2 MIN READ

Related Expertise

Authors: Joyce M. Bernasek, Matthew Quintieri

- Banking and Financial Services
- Financial Services

Background

The European Bank Recovery and Resolution Directive 2014/59 (the Directive) came into force on January 1, 2016. The Directive is designed to avoid a taxpayer-funded bailout of EU financial institutions by giving EU regulators the ability to recapitalize a failing financial institution through new bail-in powers. These new bail-in powers include writing down, converting, amending or even cancelling the liabilities of an EU financial institution to the borrower.

Parties to an agreement with EU financial institutions must include contractual terms recognizing and agreeing to be bound by these new bail-in powers if the agreement is governed by non-EU law (contractual recognition rules).

Which transactions do the new contractual recognition rules apply to?

Agreements must include contractual recognition rules if:

- the agreement is governed by non-EU law;
- an EU financial institution is a party to the agreement in any capacity; and
- the EU financial institution has potential liability under the agreement.

What are the practical implications?

The EU bail-in provisions are common in U.S. deals and are now more commonly being included in Canadian deals, regardless of any EU connection. Lenders want the provisions included to provide flexibility when assigning their commitments in the future. In addition, given that there are significant sanctions for non-compliance, borrowers should not expect to be able to negotiate these provisions. The Loan Syndications and Trading Association (LSTA) has published model clauses in relation to lender liability under credit agreements and EU bail-in clauses are now included in those provisions. Most credit agreements in Canada and the U.S. are following the LSTA model.

If you have any questions regarding these new bail-in rules and wish to discuss further,



